PREDICTING THE MOTION TRAJECTORIES OF A MODULAR SNAKE ROBOT PERFORMING VARIOUS GAITS

Cid Gilani

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING
IN
MECHANICAL ENGINEERING
AT THE
UNIVERSITY OF CANTERBURY,
CHRISTCHURCH, NEW ZEALAND.

October, 2017

Abstract

Some studies have demonstrated that snake robots have more advantages than other mobile robots in real-world applications such as search-and-rescue operations and building inspections. A pitch-and-yaw modular snake robot is capable of performing various gaits such as linear progression, rolling, sidewinding and rotating. These gaits allow the robot locomotion to adapt to different types of terrain. Predicting the motion trajectories of the gaits is necessary in order to control the robots in path planning.

This thesis presents the pitch-and-yaw modular snake robot developed as a robotic platform to study snake robot locomotion. The joint control of the robot was optimised and optimal gait parameters of the sinusoid function generators, which are used to generate the gaits, were obtained. The reprojection errors of the motion capture, which was used to track the robot's joints, were determined.

The accuracy of the 2D and the 3D simple kinematic motion models, which are based on the *virtual chassis*, were analysed using the motion trajectories recorded from the laboratory and the ODE simulation. The models were optimised to find the average optimal ground parameters of the models by training the models with the experimentally recorded motion trajectories. The consistency in the motion trajectories for each gait was evaluated to determine the predictability of each gait. The results are presented in terms of errors in the travelled distance and direction, and the orientation of the robot. The 2D model predicted the motion trajectories of the linear progression gait and the turning gait with highest average errors of 5.4% and 10.1%. The 3D model achieved similar outcomes for the rolling gait and the rotating gait but with highest average errors of 16.5% and 10.2%. However, the failure of both the 3D model and the simulation to predict the sidewinding gait might be due to the large inconsistency in the motion trajectories of the gait and the lacking number of modules. The use of average optimal ground parameters, reprojection errors and inconsistency in the motion trajectories might have contributed to the errors of the models.

Keywords: gait parameters, ODE engine, virtual chassis, kinematic motion models

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Deputy Vice-Chancellor's Office
Postgraduate Office

Co-Authorship Form

This form is to accompany the submission of any thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication. A copy of this form should be included for each co-authored work that is included in the thesis. Completed forms should be included at the front (after the thesis abstract) of each copy of the thesis submitted for examination and library deposit.

Unpublished Handbook: M.J. Koopaee, C. Gilani, C. Scott, X. Chen, "Bio Inspired Snake Robots: Design, Modelling and Control," in *Handbook of Research on Biomimetics and Biomedical Robotics*, Cairo: IGI Global, 2018.

Components of thesis extracted from Publication 1:

Chapter 3, particularly section 3.3

Conference Paper: C. Gilani, X. Chen, C. Pretty, and C. Koike, "Visualisation of the Motion Trajectory for Rolling Motion of Snake Robots using Virtual Chassis and Simplified Kinematics Motion Model," *IFAC-PapersOnLine,* vol. 49, no. 32, 2016

Components of thesis extracted from Publication 2:

Chapter 6, particularly sections 6.2-6.9

Unpublished Handbook:

The contents of the chapter were supervised by Prof. XiaoQi and edited by Mohammadali Javaheri Koopaee. The student contributed particularly on the mechanical design of a snake robot. Other co-authors prepared other contents of the chapter.

Conference Paper:

Method developed through discussions with Prof. XiaoQi and Dr. Chris Pretty. Results, written content and figures are produced by the student. All co-authors provided comments and review

Certification by Co-authors:

If there is more than one co-author then a single co-author can sign on behalf of all

The undersigned certifys that:

- The above statement correctly reflects the nature and extent of the PhD candidate's contribution to this co-authored work
- In cases where the candidate was the lead author of the co-authored work he or she wrote the text

Name: *XiaoQi Chen* Signature: Chens Date: *28/09/2017*

Preface

This thesis was conducted at the Department of Mechanical Engineering at the University of Canterbury under the supervision of Prof. XiaoQi Chen. It involves the design and development of the first prototype of a modular snake robot at the university and the study on the locomotion of the snake robot.

I would like to acknowledge and thank many people who directly and indirectly contributed to the thesis. I would like to warmly thank my supervisor, Prof. XiaoQi Chen, for his guidance during the research and preparation of the thesis. I would also like to tremendously thank my co-supervisor, Dr. Chris Pretty, for his ideas and advice during the preparation of the thesis.

I would like to add special thanks to a visiting scholar, Assoc. Prof. Carla Koike, for introducing the research topic to me. I would like to thank all visiting research students who assisted me specifically on the design and development of the modular snake robot. It was a wonderful experience for me to meet people from Europe with different cultural backgrounds.

Thanks to everyone who worked along with me at the fourth level of the ICT building. It was nice knowing you all, and I enjoyed all the intellectual and funny discussions we had at the tearoom. Last but not least, I would like to thank my family for their financial and emotional support during my studies at the University of Canterbury.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ACM	Active Cord Mechanism
CMU	Carnegie Mellon University
CPG	Central Pattern Generator
EKF	Extended Kalman Filter
FSR	Force-Resistive sensors
SVD	Singular Value Decomposition
ABS	Acrylonitrile Butadiene Styrene
GCT	Ground Contact Track
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
ODE	Open Dynamics Engine
PLA	Polylactic Acid
ROS	Robot Operating System

Symbols

ω	The angular frequency of the pitch and the yaw function generator
ϕ	The orientation of the snake robot
ψ	The phase of the pitch and the yaw function generator

θ	The direction travelled by the snake robot
$\varphi(t)_i$	The joint angle of the i^{th} module
A	The amplitude of the pitch and the yaw function generator
i	The i^{th} module of the snake robot
M	The number of modules of a snake robot
r	The distance travelled by the snake robot

Chapter 1

Introduction

1.1 Motivation

Mobile robots have proven invaluable for inspections and search-and-rescue operations in Earthquakes and other natural disasters. The purpose of this thesis is to present a study of the design and the locomotion of a snake robot that will be useful for the above purpose.

There is now a high demand for building inspections to quickly evaluate the additional risk of buildings further collapsing after an earthquake. In dealing with such a scenario, mobile robots can be handy in assisting inspection teams, keeping them from being exposed to unnecessary hazards such as radiation and poisonous gas. Mobile robots were proven to be effective tools when the Earthquake Commission (EQC) utilised a remote-controlled rover equipped with a camera to undertake building re-inspections after the Christchurch Earthquakes [1].

Mobile robots such as modular snakes have a number of advantages over other mobile robots in post-natural disaster operations. Snake robots can reach places beyond human capability, assisting search-and-rescue teams to find victims in collapsed buildings and to provide the trapped victims with survival provisions. The following points are the key factors as to why a snake robot is the most ideal mobile robot for post-natural disaster operations [2].

1. **Compact Size**

   Since snake robots have an elongated body and small cross sections, they can easily navigate through confined spaces such as gaps and pipes. This ability

is useful for inspecting gaps in rubble which are inaccessible for other mobile robots.

2. **Versatile Locomotion**

   The locomotion of snake robots is very stable because in most cases, the movement can only be generated when their long body is constantly in contact with the ground which makes it difficult to topple them over. In addition, the orientation has a minor effect on the locomotion as the robot can adapt itself to its current orientation. A snake robot can climb over obstacles which are higher than the height of its body as well. With these abilities, the robot can be used to explore uneven terrains in earthquake-affected areas.

3. **Redundant Design**

   Since snake robots are designed with several identical joint modules connected in series, the robots can still move if one of their joints malfunctions during the deployment. The identical modules are also easy to be replaced with new ones when they are faulty. These abilities are vital for time-critical operations such as search-and-rescue.

The occurrences of major earthquakes such as those in Christchurch and Kaikoura provided the motivation to develop a modular snake robot and to investigate further on its locomotion as more earthquakes are expected to occur in New Zealand in the future.

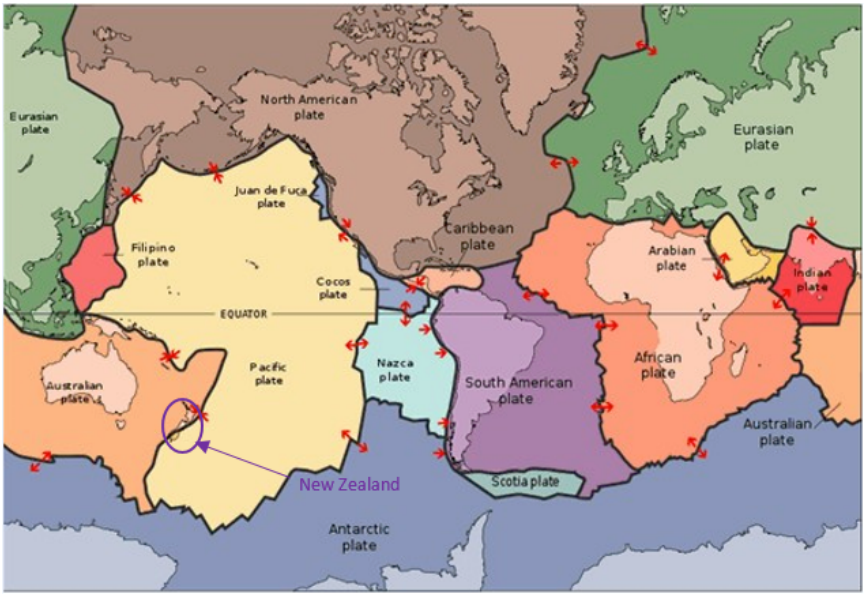


Figure 1.1: New Zealand is situated between the Australian Plate and the Pacific Plate (adapted from [3]).

According to GNS Science, approximately 15,000 earthquakes occur within and around

New Zealand every year, but only about 100-150 of them are strong enough to be felt and the rest of them are detected from the records generated by seismographs. The high frequency of this phenomenon can be explained by Figure 1.1 showing that New Zealand is located between two tectonic plates: the Australian Plate and the Pacific Plate [3].

Two major earthquakes occurred in Christchurch within two consecutive years. The first earthquake took place in September 2010 followed by the second one in February 2011. The second earthquake caused significant loss of life and damages to the buildings and infrastructure in the city which the first earthquake had already weakened. Last year, there was another major earthquake that occurred in Kaikoura.

Figure 1.2: Devastated building at Manchester Street, Christchurch (captured by Martin Hunter/Getty Images and retrieved from [4]).

The earthquake not only caused the buildings and the infrastructure in Kaikoura to collapse, but there were also significant changes made to the geographical landscape of the town [5]. Wellington (which is located north of Kaikoura) was also affected by the earthquake with some damages to buildings which required inspections to be undertaken before residents could reoccupy them [6].

The primary goal of this thesis is to predict motion trajectories of a modular snake robot performing various gaits on a planar surface using simple kinematic motion models. The gaits, which are useful for the robot to gain reliable access for building inspections and search-and-rescue operations, are investigated. Motion analysis on the motion trajectories is performed to determine the accuracy of the motion models. The next section covers the objectives that need to be accomplished in order to meet this goals.

1.2 Objectives

To meet the primary goal of this thesis, a robust snake robot must be developed as a robotic platform. A reliable software, which controls the robot with an intuitive user interface, is necessary for the operator. Predicting the motion trajectories of the gaits that can be performed by the snake robot is necessary to control the robot in path planning. The primary objectives of this thesis are listed below.

A. Develop a Reliable and Robust Modular Snake Robot

A reliable and robust modular snake robot inspired by previous studies should be designed and manufactured. In the electronic design, it is important to ensure that the robot reliably sends and receives data from a computer for data logging. A reliable software is required to control the locomotion of the robot. Though the mechanical and electronic designs may take several iterations, a fewer number of iterations in the designs is ideal for minimising the research budget.

B. Optimise the Locomotion Control of the Gaits

The joint control of the robot needs to be optimised to achieve stable and accurate motions of the joints. This optimisation is important to minimise the errors in the motion trajectories generated by the robot. The gait parameters of the robot need to be optimised to ensure that the robot generates predictable motion trajectories of the gaits.

C. Predict the Motion Trajectories of the Gaits

The motion capture needs to be developed to track the motion of the robot performing various gaits. The models of the gaits based on previous studies should be developed to predict the motion trajectories of the robot on a planar surface for different types of gaits. These predictions are limited to motion trajectories generated with certain sets of gait parameters, meaning the motion trajectories during gait transition and gait initialisation will not be analysed. The accuracy of the models needs to be validated using the experimental data conducted on the actual snake robot and simulated robot in the V-REP simulator.

1.3 Thesis Structure

Chapter 2

This chapter reviews the physiology and locomotion of a biological snake. It also reviews studies related to snake robot locomotion, robotic motion tracking and snake robot design.

Chapter 3

This chapter presents the mechanical design of the modular snake robot and the results of the strength analysis of the robot's joints. Limitation of the electronics and electrical system of the robot are discussed. This chapter also covers features of the software developed to control the robot's locomotion.

Chapter 4

This chapter covers the performance of the optimal joint control and presents optimal gait parameters to perform specific gaits for the snake robot in the V-REP simulator and during the physical trials of the robot.

Chapter 5

This chapter introduces the approaches taken to capture the motions of the robot's joints. The results of the motion capture are presented and their accuracy is determined.

Chapter 6

This chapter introduces two simple kinematic motion models used to predict the motion trajectories. The procedures taken to determine the accuracy of the models are described. Their accuracy is presented in terms of the errors in the distance, direction and orientation travelled by the robot.

Chapter 7

This chapter concludes the thesis and discusses the future work that can be undertaken to improve and to extend the research on the modular snake robot.

Appendices

Additional data and results from some chapters of the thesis are presented as references in Appendices A-G.

Chapter 2

Literature Review

2.1 Biological Snake

It is paramount to have a general understanding of the physiology and locomotion of a biological snake before discussing design, control and modelling of a snake robot. Snakes are vertebrates with between 100-400 vertebrae and each intervertebral joint can rotate horizontally and vertically about 10-20° and 2-3° respectively. The vertebrae of snakes can be divided into four main sections as shown in Figure 2.1 with each section consisting of distinctive shape of vertebrae [7].

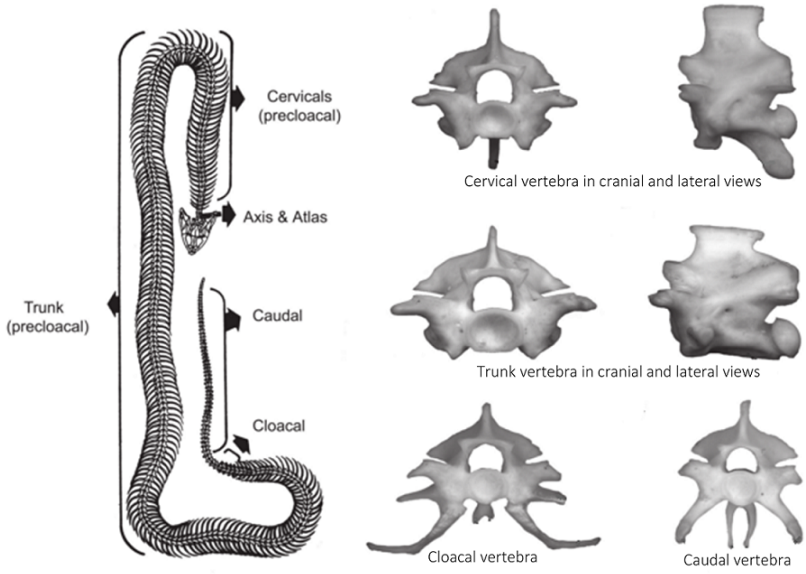


Figure 2.1: The vertebrae of the snake can be be divided into cervicals, trunk, cloacal and caudal sections (adapted from [7]).

There are four types of locomotion observed in snakes [8]. The characteristics of each gait are presented in Figure 2.2. The following are the types of locomotion of snakes:

A. Lateral Undulation

This gait, also known as serpentine, is the most common and fastest gait that is generated by continuous wave propagation in the backwards direction along the snake's body from head to tail. It relies heavily on dynamic friction between the side of the snake's body and an uneven surface, so this gait is not suitable for locomotion on a slippery surface. During this gait, the snake slightly lifts the peaks of its body wave curvature from the ground to distribute the body weight along its body curve.

B. Concertina

The snake commonly utilises this gait to move in narrow passages or along a branch on a tree. The gait is generated by extending the front part of the body while the back part is curled into several waves to provide an anchor against the constricted environment.

C. Rectilinear

This gait, also known as caterpillar gait, is the slowest gait in snake locomotion and is commonly utilised by large-sized snakes. The gait is generated by stretching forward some parts of the body and hooking them over the ground using the edges of the scales. As parts anchored to the ground start pulling the rest of the body, the snake generates forward movement. The step is repeated several times using other parts of the body. This gait utilises the muscles which control the scales of the snake to anchor the body to the ground and provide stretching and pull actions.

D. Sidewinding

Snakes in the desert uses this gait to travel on the sand. It resembles concertina, but the gait requires only two points of contact. Motion generated by throwing the head and the neck to the sideways while the rest of the body provides an anchor to the ground. The head and the neck, in turn, act as an anchor while the rest of the body repeats the similar motion. This gait moves 45° relative to the direction of its body length.

In snake locomotion, the snake does not switch its gait immediately. Instead, it switches to a new gait as its speed of movement increases. The gait preferred by the snake typically depends on its body type and surrounding terrain condition. The size of the snake is also another factor that determines the snake's preference in gait for locomotion. For instance, a small snake will likely use lateral undulation gait whereas a large-sized snake tends to use rectilinear gait [9].

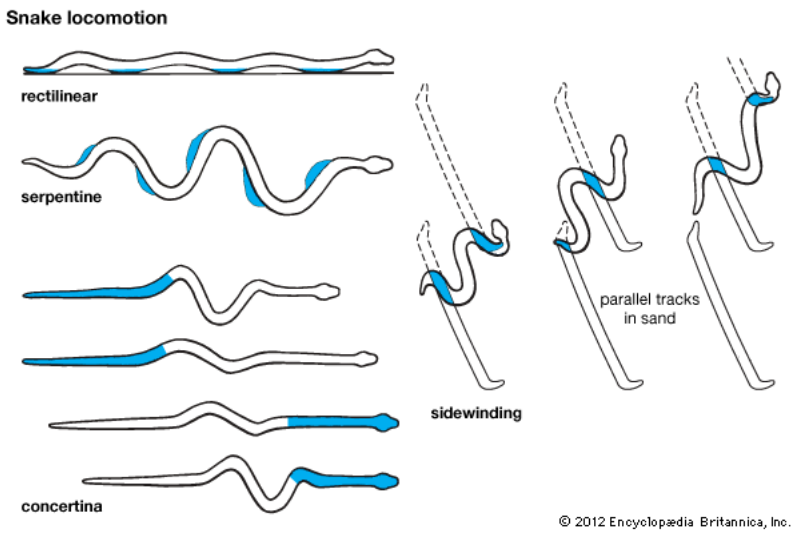


Figure 2.2: The ground contact points of the snake while performing the linear progression, concertina, rectilinear and sidewinding gaits (retrieved from [10]).

The scales of a snake are vital for lateral undulation and rectilinear gaits. The scales provide anisotropic friction property between the snake's body and terrestrial surface. As shown in Figure 2.3b, the anisotropic friction is generated because the static friction at the flank of a snake is greater compared to any other direction of the snake body. In addition, the scales of a snake are stacked upon one another in the direction from its head to its tail (the front scale is always on the top of the back scale as shown in Figure 2.3a) that make it impossible for the snake to move backwards [11]. Furthermore, a snake is capable of changing the direction of the friction of its belly by bending its body which causes its scales to fan outward radially as shown in Figure 2.3b [9]. The static friction at the outer bending curve may likely be higher than at the inner bending curve because the scales are far more lifted at the outer curve. This capability helps a snake to slither and to hook onto the ground while performing lateral undulation gait and rectilinear gait respectively.

(a) **(b)**

Figure 2.3: The scales on the belly of a snake (a) and when its body is bent (b) (retrieved from [9]).

2.2 Snake Robot Locomotion

2.2.1 Gaits

The capability of a snake robot to perform gaits depends on how the robot is designed. There are various practical gaits that have been developed for snake robots. According to Lipkin et al. [12], the gaits can be divided into two type of gaits: biologically inspired and non-biologically inspired gaits. The biologically inspired gaits, such as lateral undulation, rectilinear and sidewinding gaits are the most common studied gaits in snake robot locomotion. These gaits are not implemented in the same way as biological snakes because snake robots do not have the muscles and scales which play an important part in biological snake locomotion. As for non-biological snake gaits, rotating (turn-in-place) and rolling are two common gaits found to be useful in many snake locomotion studies.

Similar to a biological snake, a lateral undulation is the fastest gait in snake robot locomotion but it requires anisotropic friction between the robot's body and the ground. This friction constraints a robot's joints from slipping to the side. It can be achieved by equipping snake robots with passive wheels underneath them. This gait is also applicable for snake robots swimming at the water's surface as presented by Crespi et al. [13]. Lateral undulation gait can be slightly modified to perform sinus-lifting by adding a vertical wave motion. Sinus-lifting is effective to allow the robot to travel on both slippery or roughed floor. It enables the robot to manipulate the friction between its body and the floor by controlling the amount of pressure exerted over a few wheels [14].

A linear progression gait, also known as inchworm gait, resembles the concertina and rectilinear gaits of a biological snake and is a very useful gait for precise control of the robot's position [15]. As compared to an actual concertina gait, this gait is believed to be better because in a concertina gait, the momentum is not conserved efficiently and the pressure is highly concentrated over a small number of joints [12]. This gait is useful for a snake robot to travel in confined spaces such as inside pipes and under gaps which require high precision positioning. Generating the gait waveform in both the horizontal and vertical axes enables the snake robot to climb a wider pipe and to increase its climbing speed [12]. For optimal climbing, the presence of a complaint material is vital to ensure the robot's body has adequate contact with the wall to be climbed [16].

A snake robot can navigate around a corner with a turning gait which is generated when the robot curls its body into an arc shape while performing the linear progression

gait. The robots will travel along a circular path formed by the radius of the arc. However, this gait is not suitable for navigating through a sharp corner like a pipe so generating a sharp bend would be the best option. This is done by having an arc shape of small diameter produced at the front and then shifted to the back modules as the front modules are straightened while leaving the corner.

Sidewinding gait generates lateral movement and can be modified into conical sidewinding gait which can be used to steer the robot in a similar manner to a differential-drive car [17]. This steering motion is achieved by forming the robot into a cone shape while performing sidewinding [18]. Hatton and Choset [19] demonstrated a modified version of sidewinding gait that can climb on a slope. A further study was conducted to extend the capability of the sidewinding gait to climb on a sandy slope by mimicking the motion of a rattlesnake [20].

The non-biological inspired gait, the rotating gait, is very similar to the sidewinding gait. Unlike the sidewinding gait, however, the robot's body rotates about its centroid when performing rotating gait [21, 22]. The main difference in the movement characteristics is caused by the ratio of the phase differences between the joints in the vertical and the horizontal axes [21]. This gait is useful when there is limited space for the robot to perform the turning gait.

A lateral rolling gait is the most efficient gait and it can be modified into different types of rolling gaits to climb on a vertical pole and roll on a horizontal pole [23]. Helical rolling gait is one of the forms of the rolling gait that enables a snake robot to climb a vertical pole with adequate gripping. Although this gait provides more balance to the robot while travelling on a horizontal pole, the movement of the gait is not as efficient as performing lateral rolling on the pole [24]. The study [25] demonstrated that a hump could be added to the robot's shape while it is rolling to navigate through obstacles. In addition, an S-shape rolling gait is also a form of rolling gait which can generate lateral and forward movements [15].

2.2.2 Modelling

Lateral undulation and linear progression gaits are considered as 2D locomotion as a robot's joints move in 2D space with a single axis during the locomotion. In contrast, turning, sidewinding, rotating, and rolling gaits are considered as 3D locomotion because the joints rotate in 3D space with joints rotating about two axes during the locomotion. Unlike lateral undulation gait, predicting motion trajectories for other gaits are challenging as the movements of a robot's joints are difficult to track and change depending on the gait control.

Most of the models of snake robots are developed for lateral undulation gait specifically on a planar surface. Before the gait can be modelled, it is important to first model the lateral undulation gait performed by biological snakes. The biological snake model studied by Hirose [26] has contributed significantly to the modelling of lateral undulation gait in snake robots. He discovered that the Serpenoid curve shown in Equation 2.1 was the closest representation of lateral undulation where $x(s)$ and $y(s)$ are the coordinate points along the curve of arc length s from the origin, and a, b and c are positive scalars.

$$\begin{aligned} x(s) &= \int_0^s sin(acos(b\alpha) + c\alpha)d\alpha \\ y(s) &= \int_0^s sin(acos(b\alpha) + c\alpha)d\alpha \end{aligned} \tag{2.1}$$

He also discovered the effects of environment on a biological snake's shape during locomotion and further developed mathematical models which consider external variables such as ground friction and temperature. However, it was later discovered by Ma [27] that Serpentine curve represents the closest approximation to the lateral undulation of a biological snake. This discovery was validated by comparing the locomotion efficiency between two other models: serpentine curve and serpenoid curve.

In general, the kinematics of snake robots can be expressed by using homogeneous transformation [28, 29]. The modelling of lateral undulation gait is simplified by applying side-slip constraints to snake robots. Side-slip constraints can be easily achieved by equipping their joints with passive wheels. Shugen [28] applied Newton–Euler equation to derive the robot dynamics, and Ma and Tadokore [30] further extended the model for locomotion on a slope. Furthermore, Ma et al. [31] modelled a snake robot performing lateral undulation gait with sinus lifting by considering the robot's mass distribution. Liljebäck et al. [32] modelled the contact force between the robot and the ground as a mass-spring-damper system which forms the third axis of the proposed model. Since these models describe the

motion trajectories of the gait, they can be used to control the position, speed and torque of the robot during the locomotion.

There are very few studies conducted to model the 3D locomotion. It is difficult to model the turning, rolling, sidewinding and rotating gaits because the snake robot's joints do not often touch the ground during the locomotion. In addition, the time when the joints touch the ground depends on the gait setting. Thus, the body frame of the robot needs to be defined to track the joints' movements. Rollison and Choset [33] proposed a method called *virtual chassis*, which separates the *internal motion* (the joints' movements) and the *external motion* (the motion trajectories) of a snake robot. Enner et al. [34] proposed a simplified motion model which is based on the virtual chassis. The model assumes that there is no slip during the locomotion and uses a pair of parameters to consider irregularity contacts between the module and the ground. The highest average of the errors of the model for rolling, sidewinding, slithering, and rotating (turn-in-place) gaits are 9%, 28%, 6%, and 34% respectively [34]. Melo [35] proposed an alternative method called "floating frame of reference" (FFR) to define the body frame of a snake robot. The velocity travelled by a robot during sidewinding gait was modelled based on FFR, but this velocity does not describe the direction and orientation of the robot. It was found that the accuracy of the model depends on the magnitude of the vertical waveform and the speed of the gait. When the magnitudes of both waveforms are equal, the error does not exceed 6.1%. The highest error is 15% when the vertical waveform has smaller amplitude than the horizontal waveform. The model becomes less accurate when the speed of the gait increases and is invalid when the wavelength of the gait decreases [35].

2.2.3 Control

The control of snake robots is complex due to the great number of degrees of freedom of snake robots with less independent input controls. There are two types of control in snake robots' locomotion: closed-loop control and open-loop control. Closed-loop control such as model-based control focuses on controlling the path of the robot locomotion. Another form of a closed-loop control is the obstacle-aided control that utilises obstacles for propulsion instead of avoiding them. An open-loop control such as Central Pattern Generator (CPG) control generates rhythm patterns of the robot locomotive motion.

Model-based controls are common for controlling the motion trajectory of lateral undulation gait. The model-based control proposed by Prautsch [36] directs the position and the 'head' (first) module trace of a snake robot using a PD control which is based on

Lyapunov function. The control ensures that the robot performs a smooth and efficient lateral undulation motion while maintaining its head module trace in Serpenoid pattern. Liljebäck proposed a motion model-based control to direct snake robots in a straight line path [37]. The stability of the control is validated by analysing it using a Poincare map. The study also found that a snake robot is only controllable on anisotropic ground friction not on isotropic ground friction.

There are two main approaches proposed in obstacle-aided control. The use of tactile sensors is an approach that is commonly used to detect obstacles. Bayraktaroglu [38] implemented tactile sensors on the head module to explore the environment and the tactile sensors on the other modules to find three contact points with obstacles. He then used a curve fitting method accordingly to the three contact points to generate propulsion. Kano and Ishiguro [39] proposed a method for the snake robot to navigate through an unstructured environment by using the curvature derivative control that took the pressure sensors on the sides of the robot as the control inputs. The study conducted by Transeth et al. [40] proposed a non-smooth mathematical model for obstacle-aided locomotion. Unlike the previous studies that rely on tactile sensor, the closed-loop curvature control proposed by Andruska and Peterson [41] enables the snake robot to navigate through unknown curving paths with current sensing only.

Another approach for obstacle-aided control is to utilise compliant behaviour in snake robots to interact with objects in the surroundings. The compliant behaviour not only allows snake robots to use obstacles in locomotion, but also enables it to grip onto objects. Torque control strategies were proposed by Rollinson et al. [42] to enable the SEA snake to adapt its body shape to its surroundings during locomotion. Rollison and Choset [43] managed to generate autonomous complaint behaviour in snake robots' joints by using the Extended Kalman Filter (EKF) without torque sensors and mechanical compliance.

In nature, the CPG plays a significant role in generating rhythm patterns of locomotion of animals such as flying, swimming, walking and crawling without the need of any sensory signal. Also, the shape and speed of the cyclic movement can also be altered in the CPG [44]. A CPG-based control is commonly implemented in snake robot locomotion as illustrated in Figure 2.4. The CPG-based control proposed by Inoue et. al [45] is an open-loop control system and it was further developed by Wu and Ma [46] to receive feedback as illustrated in Figure 2.5. To avoid obstacles, Wu and Ma [47] then included two more control approaches into the control system of their robot: head-navigated and turn motion

controls. Furthermore, a CPG-based control receiving feedback from friction force sensors is proposed by Inoue et. al [48] to enable a snake robot to adapt to the change of ground friction by using Ekberg's model and Matsuoka's model.

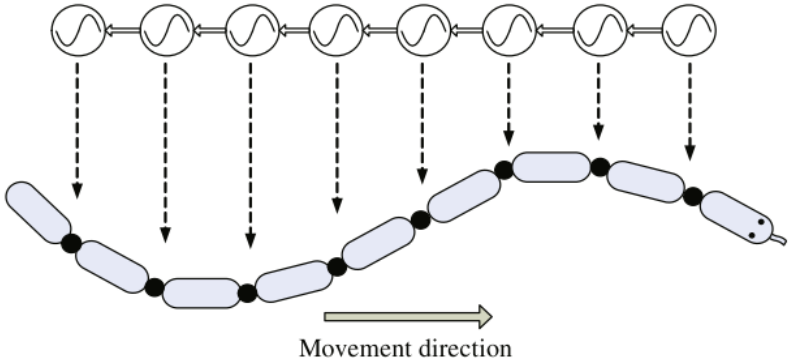


Figure 2.4: The implementation of CPG-based control in snake robots (adapted from [46].

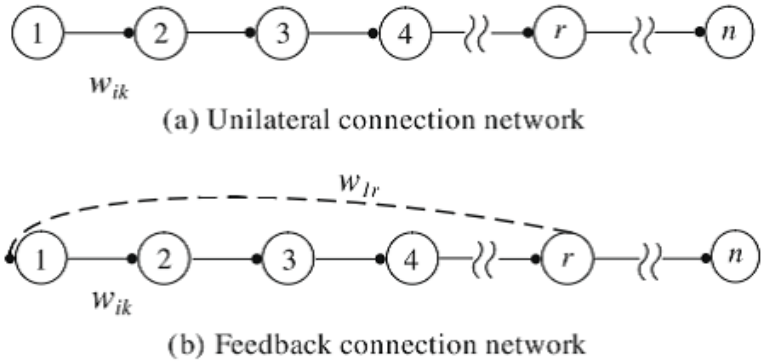


Figure 2.5: Schematics of two distinct CPG connection networks (adapted from [46]).

The sinusoidal function generator control in Equation 2.2 is very similar to CPG-based control but with a simpler parameter control that changes only the amplitude, frequency and phase shift of the generated sinusoidal waveform. The notations of the equation is defined in Table 2.1.

$$\varphi(t)_i = A_i sin(\omega_i t + \psi_i) + O_i \quad i \in 1....M \tag{2.2}$$

Table 2.1: Definition and range of notations found in the sinusoidal function generator.

Notations	Definition	Range
$\varphi(t)_i$	Joint angle of the i$^{\text{th}}$ module	[-90, 90] degrees
A_i	Amplitude of the i$^{\text{th}}$ function generator	[0, 90] degrees
ψ_i	Phase of the i$^{\text{th}}$ function generator	(-180, 180] degrees
ω	Angular frequency of the i$^{\text{th}}$ function generator	$\omega \geq 0$ rad/s
O_i	Offset of the i$^{\text{th}}$ function generator	[-90, 90] degrees
M	Number of modules of the snake robot	$M \geq 2$

For a modular snake robot with pitch-and-yaw modules, the function generator controller is slightly modified to form two sets of function generators as shown in Equation 2.3 [22, 12]: the pitch and the yaw function generators. The pitch function generator controls the pitch modules that rotate in the horizontal axis whereas the yaw function generator controls the yaw modules that rotates about the vertical axis.

$$\begin{aligned} \varphi(t)_{i,p} &= A_p \sin(\omega_p t + \psi_p(i-1)) \quad i \in \{1...M_p\} \\ \varphi(t)_{i,y} &= A_y \sin(\omega_y t + \psi_y(i-1) + \psi_{py}) + O_y \quad i \in \{1...M_y\} \end{aligned} \tag{2.3}$$

where i is the module number of the robot, $\varphi(t)_{i,p}$ and $\varphi(t)_{i,y}$ are the rotation angles of pitch and yaw modules respectively, ω_p and ω_y controls the speed of the locomotion speed, and ψ_p and ψ_y controls the shape of the snake robot. An additional parameter ψ_{py} is added in the yaw generator to set the phase difference between the yaw and the pitch generators. O_y is the constant angle added to yaw function generator. The subscripts p and y indicate that the parameters in the pitch and the yaw functions are independent. Table 2.2 presents the specific gait parameters required to generate linear progression, turning, sidewinding, rotating, lateral rolling, and helix rolling gaits [22, 12, 24].

Table 2.2: The specific gait parameters required to generate different type of gaits .

Gait	Gait Parameters
Linear Progression	$A_p > 0°$, $\psi_p > 0°$
Turning	$A_p > 0°$, $\psi_p > 0°$, $O_y \neq 0°$
Sidewinding	$\psi_p = \psi_y$
Rotating	$\frac{\psi_p}{\psi_y} = 0.5\text{-}0.6$
Lateral Rolling	$A_p = A_y$, $\psi_{py} = 90°$
Helix Rolling	$A_p = A_y$, $\psi_p = \psi_y$, $\psi_p > 0°$, $\psi_{py} = 90°$

The function generators can be implemented together with the annealed chain fitting method that enables the shape of the robot to fit to a backbone curve [49]. This method can be used to optimised the shape of the robot so that it will perform gaits more effectively.

Since the gaits generated using the controller are not closed-loop controlled, Hatton et al. [50] proposed a path-following method that can be used to direct the robot to a target position using various gaits. The approach utilises the virtual chassis that is used to indicate the gaits required to move in different directions as illustrated in Figure 2.6. A closed-loop control was later developed to direct the snake to follow a trajectory path [17]. The study proposed a control method to steer the sidewinding gait in a similar manner to a differential drive vehicle.

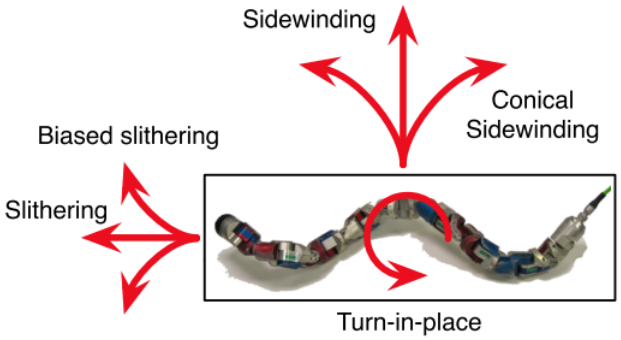


Figure 2.6: The abstraction of the motion of a snake robot in multiple directions using various gaits (retrieved from [50]).

2.3 Snake Robot Design

Biological snakes highly inspire the design of snake robots. Snake robots are designed with several segments driven by actuators which rotate horizontally or vertically (1-DOF rotation), or both (2-DOF rotation) to imitate the vertebrae of biological snakes. According to the survey [2], snake robot designs can be classified into five general categories based on locomotive-assisting mechanism. They are snake robots with passive wheels (Type I), active wheels (Type II), active treads (Type III), 3D joints (Type IV), and linear expansion joint (Type V). Figure 2.7 presents an example of a snake robot belonging to each of the five categories.

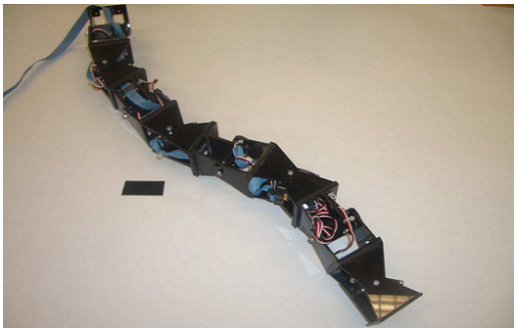
The survey concluded with three general observations. First, snake robots with longer length tend to move faster because they usually consist of more joints, which subsequently

generate a larger propulsive force. Secondly, the rectilinear gait performed by Type IV and Type V was generally slower than the lateral undulation gait performed by Type I-III. Nevertheless, the surveyors later on developed a snake robot specialising in a rectilinear gait that outperformed most of the snake robots in the survey by being the second fastest snake robot [51]. Lastly, it is not fair to compare the designs of snake robots based on a single criterion. For instance, AmphiBot II [52] may travel faster than CMU snake robots [53], but could only perform a single gait [2].

(a) Type I: Amphibot I (retrieved from [8])

(b) Type II: GMD-Snake2 (retrieved from [54])

(c) Type III: OmniTread OT-4 (retrieved from [55]).



(d) Type IV: 'Y1' Modular snake robot (retrieved from [21]).

(e) Type V: Slim Slime (retrieved from ([56])

Figure 2.7: Snake robots with (a) passive wheels, (b) active wheels, active threads, (d) 3D joints, and (e) linear expansion mechanism.

Furthermore, most of the snake robots in the survey are designed as modular robots consisting of a single chain of modules (1D Topology) [22]. The advantages of modularity in snake robot design are that it enables the snake robot's length to be extended, it lowers the manufacturing cost and it is easier to repair [57]. The following is the ideal design specification of a snake robot: it should possess a small body diameter, high-speed movement, should be able to perform various gaits, and its batteries can last over a reasonably long duration [2]. The requirements to meet these criteria depends on the application of a snake robot. Some applications require the robot to have mechanical and electronic hardware. This increases the size and the weight of the robot. So, the requirements need to be lowered; otherwise, the cost of developing the snake robot will increase significantly. Thus, it is important to consider the application of the robot in order to set the design requirements to develop the robot.

2.3.1 Module Design

The most challenging task of designing a module for a snake robot is to optimise the diameter, length and weight of the module. The diameter and the length of a module limit a snake robot's movement in confined spaces while the weight of a module affects the power required to move and hold the snake joints. As illustrated in Figure 2.8, the capability of a snake robot to manoeuvre in confined spaces can be measured by calculating the minimum bending radius, R_{min} using Equation 2.4 where l is the length of module, and θ_{max} is the maximum rotation angle of the robot's joint [58].

$$R_{min} = \frac{l}{2\tan(\theta_{max}/2)} \tag{2.4}$$

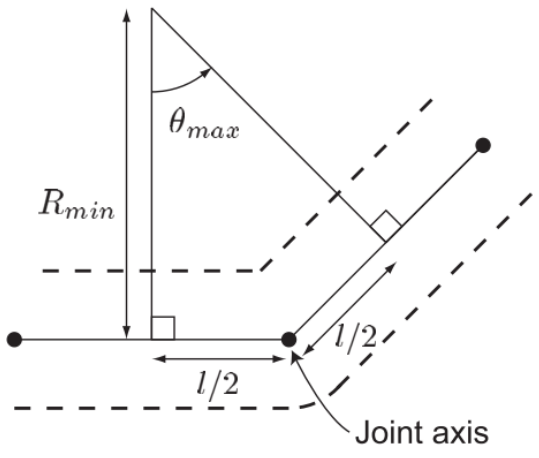


Figure 2.8: The minimum radius which a snake robot can bend in confined spaces [58].

Several design features, such as the joint types, the size of the electrical and electronic compartment and the presence of the locomotive-assisting mechanism lead to changes in R_{min}. The joint types of a snake robot are 1-DOF rotation joint and 2-DOF rotation joints. Most of the snake robots with 2-DOF rotation joint in prior studies [59, 40, 60, 61] have shorter l (decreasing R_{min}), but smaller θ_{max} (increasing R_{min}).

In contrast, the snake robots with 1-DOF rotation joints in prior studies [62, 63, 53, 64, 65] simplify the mechanical design of the module and increase θ_{max} but require twice the number of joints to achieve 2-DOF rotation. Fitting the electronic board into a module is not a major issue because it can be customised into smaller sizes to fit into the electronic compartment, but this is not the case for actuators. A module needs to be designed with adequate space to accommodate an actuator and for its gearing mechanism to fit. Locomotive-assisting components such as passive or active wheels also increase the diameter of a module and subsequently decrease θ_{max}.

There are two main factors of mechanical design which can affect the locomotion of a snake robot. First, it is important to consider the impact of mechanical components such as passive wheels on the robot that prevent it from performing complex gaits such as pole climbing and wall climbing. In addition, exploring outdoor terrains would be a difficult task with the presence of passive wheels [57]. Although Kamegawa et al. [66] has demonstrated the success of a snake robot with passive wheels climbing a pole, the use of soft urethane foam and the tyres surrounding the wheels may have significantly contributed to its success. Secondly, the shape of a module can influence the stability of a snake robot's body. A snake robot with a square shaped module may seem to be much more balanced during linear progression gait, but may not perform rolling motion smoothly. On the other hand, a snake robot with a round shape module may lose its balance during linear progression gait, but curving its body into S-shape would make it more stable [12].

2.3.2 Electrical and Electronic Design

The electrical and electronic system of a snake robot can be designed to various degrees of modularity. Ideally, the more modular the system, the more robust the electrical and electronic system is to faults. There are several modular design aspects that need to be considered. They are power, communication, and sensors. The hardware required for the design aspects can lead to an increase in the size of a module. Therefore, it is important to consider the hardware needed for the application of the snake robot.

2.3.2.1 Power Supply

To power a snake robot, there are two common methods which are widely applied: a portable power source or tethered. The method used varies depending on the type of application that the given robot is designed for. Snake robots powered by portable power supplies such as batteries are practical for exploration in the field and are usually designed with batteries in each module. Good examples of these snake robots can be found in prior studies [52, 8, 67]. In addition, portable sources of compressed air were used for the pneumatic system of OmniTread OT-4 [55]. According to Hirose and Fukushima [68], the use of a tether such as "hyper-tether" (flexible electrical cable controlled by a winch) is more practical in real world applications. The use of this tethering helps to reduce the size of a module, to establish a highly reliable communication and can be used to recover a robot when it is out of service during a deployment. The use of an ordinary tether is also common in snake robots, but the applications are usually limited to laboratory operations only. A tether can also supply water via a water tube as demonstrated by the snake robot designed by Liljebäck et al. [69].

2.3.2.2 Communication

Snake robots are commonly designed to be controlled either through wireless or wired communication. Snake robots with wireless communication will usually have a master microcontroller within the robots. The master microcontroller receives control inputs from the external computer through a wireless communication board, then performs computation and transmits output data to a microcontroller in each module of the snake robot via I^2C, TTL serial, or CAN bus communication [52, 62, 70]. However, the reliability of wireless communication depends on the operating environment of the robot, and high data transmission may cause latency in the communication with a low-end communication system. This issue makes wired tethering the best option for a highly reliable communication. Due to the implementation of RS485 and Ethernet communication, snake robots do not necessarily require a master control within the robot, and instead an external computer directly controls each module of the snake robots [64, 45].

2.3.2.3 Sensors and instrumentation

Sensors and instrumentation of a snake robot are essential for a snake robot to operate in an uncertain environment with uneven surface and obstacles, and for path planning.

The type of sensors and instrumentation used for a snake robot can be divided into several categories. These are: visual sensing, force sensing, pose sensing, local positioning, and global positioning.

Visually, a camera, which is commonly installed at the head module of a snake robot, is very vital for operators to inspect the surroundings of the robots. The information received from the camera allows the snake to detect shapes and colours. It was found that installing a camera at a second position, the middle module of the snake robot, can be very useful in gathering relevant image data for various types of gaits [71]. The cameras can either look straight ahead or to the side. Since the information of the camera is limited to only the shapes and colours, as shown in Figure 2.9, a lidar sensor was developed for the head module of the CMU modular snake robot to obtain the information of the robot's surroundings in the form of 3D point clouds [72].

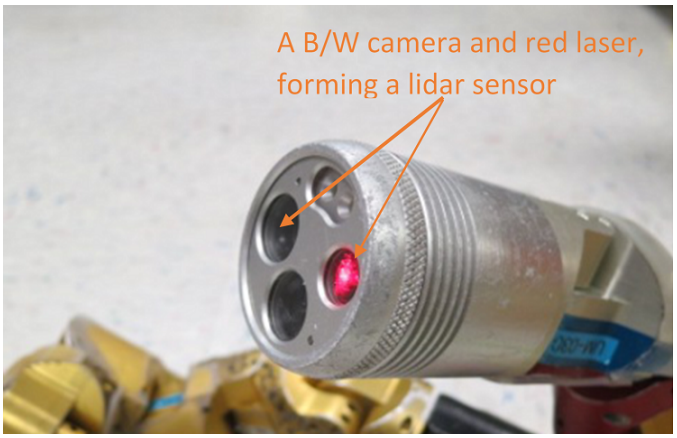
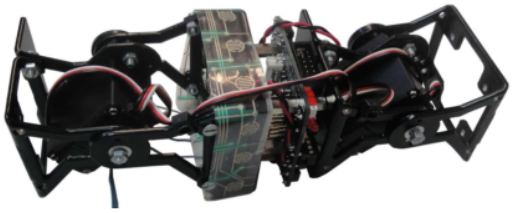
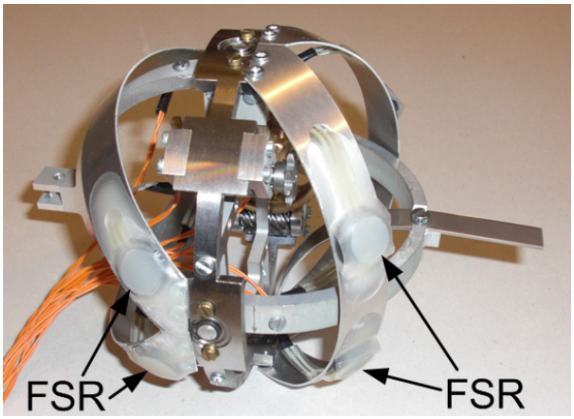


Figure 2.9: The lidar sensor in the CMU snake robot (retrieved from [72]).

It is important for snake robots to physically sense their surroundings such as objects and terrain so that they can navigate effectively as well as climb and grasp objects. There are two main methods to achieve this capability: body contact and joint contact. The body contact method detects the position of an obstacle when the readings of the force/pressure sensors on the robot's body reach at certain threshold values. One example of body contact method is the use of touch rings consisting of capacitive sensors as shown in Figure 2.10a [73]. In addition, installing pressure sensors or limit switches on the side of the snake robot's modules enables it to navigate through obstacles, but this implementation was limited for obstacles obstructing the sides of the robot [38, 39]. The implementation of force-resistive sensors (FSRs) around circular modules of the snake robot as shown in Figure 2.10b enabled it to roughly read the magnitude of the forces acting on any direction of the modules [74].



(a) A touch ring strip was wrapped around the Cube MY1 module (retrieved from [73]).



(b) FSR were installed around the circular module designed (retrieved from [74]).

Figure 2.10: The implementation of body contact methods to detect snake robots' surroundings.

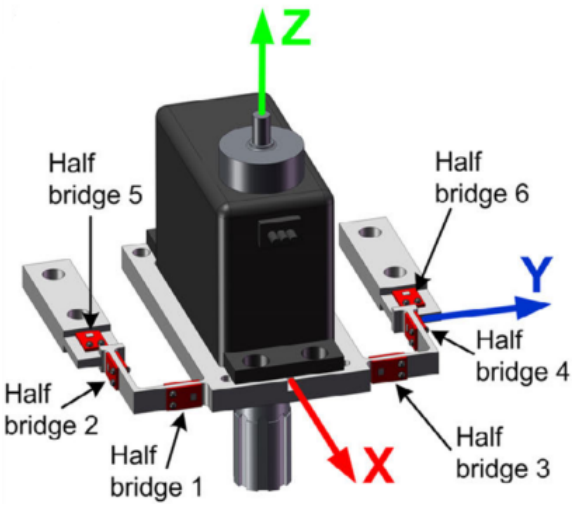


Figure 2.11: A CAD model showing the locations where the strain gauges were installed on the aluminium bracket(retrieved from [70]).

On the other hand, the joint contact method reads the torque readings exerted to the joints of a snake robot and determines the position of obstacles using that information. For instance, strain gauges can be installed on an aluminium bracket in a way that the magnitude and direction of the torque exerted to the joint of the robot can be estimated [70]. The torques can be determined using the current reading of the robot's actuators and the linear model of the series elastic joint of a module [65]. Figure 2.11 shows positions where the strain gauges were installed to achieve 3D torque reading.

Inertial sensors such as accelerometers and gyroscopes are necessary for a snake robot to track its joints orientation especially when travelling on an uneven surface, climbing a pole or on an obstacle and scanning the environment with visual sensors. The orientation of the snake robot was estimated by Rollinson et al. [75] using Extended Kalman Filter (EKF) which fused the data obtained from accelerometers, gyroscopes and joint angles of the robot.

There has been a limited amount of research conducted on localisation for snake robots. The most common method of localisation for snake robots is to track them using cameras mounted to the ceiling. However, this approach is limited to lab environment [76, 50, 17]. According to Xiao et al. [17], it is still possible to track the robots with surveillance cameras if they are available in the areas where the robots are deployed. The most reliable method would be to map the environment using onboard lidar as demonstrated by Ponte et al. [72].

2.4 Snake Robots in Other Research Institutes

2.4.1 CMU's Snake Robots

The first snake robot developed by the Carnegie Mellon University (CMU) was built with laser cut plastic modules with 1-DOF rotation joints connected in series. CMU further strengthened the design by using a bent CNC milled metal sheet. The modules were assembled in pitch-yaw-pitch order to achieve movement in 3D space. The range of rotation angle for the module was also increased to $\pm 90^{\circ}$ to improve the multi-gait functionality of the robot. The first six CMU's prototypes as shown in Figure 2.12 used modified hobby servos consisting of controller boards (Atmel microcontroller and H-bride), sensor boards (a magnetic encoder, accelerometer and gyroscope) and a power board (switching power supply). As illustrated in Figure 2.13, the connection between two modules is made by using a U-case which is attached to the output shaft of the servo at the top of the first

module, and the next module is then screwed to the lower part [53, 57].

Figure 2.12: The six snake robot prototypes showing the progression of the robot design where the newer robots are at the top of the image (retrieved from [53]).

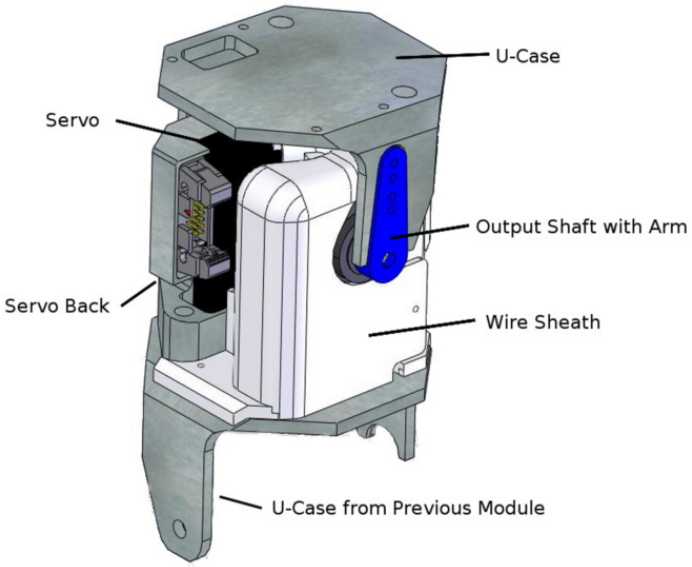


Figure 2.13: The connectors between two consecutive modules of CMU snake robots (retrieved from [57]).

The two latest prototypes, the Unified and the SEA snake robots, did not use modified hobby servos as the actuators. Instead of hobby servos, brushless DC motors were used in these robots. The electronic designs for the two prototypes were slightly modified to accommodate the brushless DC motors. Each prototype included a unique feature. The Unified snake robot installed a brake mechanism which allowed the module to stay locked in position. This mechanism significantly minimised the energy consumption of the robot

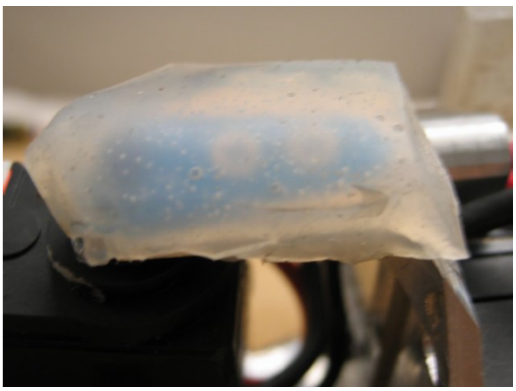
while it was gripping an object [64]. In the SEA snake robot, each module consists of a joint with compliance behaviour provided by the elastic material installed in the joint. Unlike other CMU prototypes which communicated via RS485 serial communication, the SEA snake robot communicated via Ethernet communication at a higher bit rate [65].

(a) Unified snake robot (retrieved from [64]).

(b) SEA snake robot (retrieved from [65]).

Figure 2.14: CMU snake robots with aluminium enclosed modules.

The CMU's snake robots were not powered with batteries. Instead, wired connection was used for power and communication and high-level programming performed on a PC. The designs of the head and the tail modules were different from the rest of the modules. The head module was developed with a camera and lidar, whereas the tail module was designed to hold the cable that supplied power and communication with the robot. Figure 2.15 shows the use of 'compliance' (platinum-doped silicone cover) to provide higher friction between the module and the ground surface and 'skin' underneath to protect from dirt and water [53, 57, 65, 64]. However, 'skin' was no longer necessary for the last two prototypes as they were fully enclosed with aluminium casing as shown in Figure 2.14.



(a) 'Compliance' covering a module.

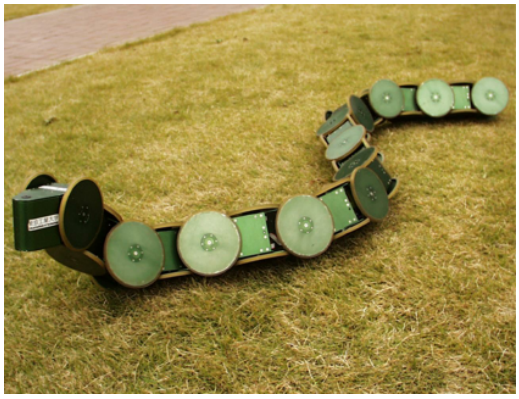
(b) 'Skin' underneath the 'compliance'.

Figure 2.15: 'Compliance' and 'skin' used in the first six CMU's prototypes (retrieved from [53]).

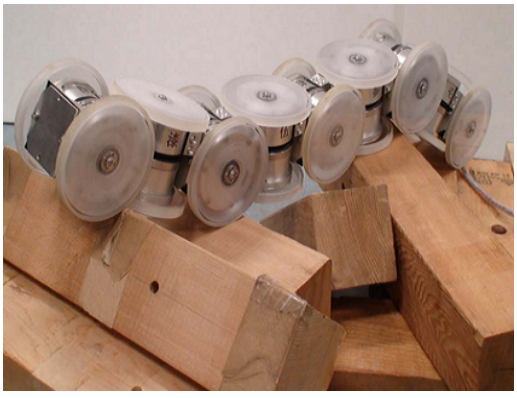
2.4.2 ACM Snake Robots

Hirose has been active in the development of Active Cord Mechanism (ACM) snake robots since the 1970's [26]. Figure 2.16 presents the commercialised ACM snake robots at HiBot. All of the ACM snake robots were equipped with passive wheels that restricted the modules to longitudinal movement only. The first snake robot equipped with two casters under each link, named the ACM-III, was developed by Hirose in 1972 to perform lateral undulation. He later developed a similar model known as the ACM-R1 which was still limited to 2D locomotion but could be controlled with a wireless remote controller. The first snake robot developed by Hirose with 3D locomotion capability was the ACM-R3. This robot consisted of modules with 1-DOF rotation joints arranged in pitch-yaw-pitch order. The robot was also equipped with large passive wheels which were able to roll against obstacles and enabled the robot to perform lateral undulation gait as well as sidewinding gait. The ACM-R4 was then developed based on the design of the ACM-R3, but was equipped with large active wheels [2].

(a) ACM-R1 (retrieved from [77])



(b) ACM-R3 (retrieved from [2])



(c) ACM-R4.1 (retrieved from [78])

(d) ACM-R5 (retrieved from [77])

Figure 2.16: The development of ACM snake robots.

Unlike the ACM-R3, the ACM-R4 was capable of rotating up to $\pm 90°$ while the wheels

were equipped [78]. The ACM-R5 was later developed specifically for an underwater environment. This robot consisted of several modules with 2-DOF rotation joints, and each module was equipped with small passive wheels which allowed it to slither on land. In addition, the fins on the modules of the robot enabled it to swim in the water [59].

Hirose and Yamada [58] later developed a miniature snake robot with 2-DOF rotation joints as shown in Figure 2.18. A method to measure the capability of a snake robot's ability to manoeuvre in confined spaces was proposed. The capability can be measured by calculating a minimum bending radius, R_{min} using Equation 2.4. Unlike the prior ACM snake robots which had $R_{min} > 100mm$, this miniature ACM model focused on minimising the R_{min} by decreasing the length and increasing the angle of rotation of the universal joint module. The new ACM model was designed based on the concept of the 2-DOF rotation joint mechanism as illustrated in Figure 2.17.

The new design managed to reduce R_{min} significantly to 73 mm with each module having a dimension of 100 mm × 39 mm × 40 mm (length × width × height). The length between two consecutive axes was 50 mm. Two motors, two potentiometers, a microcontroller and two motor drivers were required by each module, and with these components, a closed-loop positional control was formed.

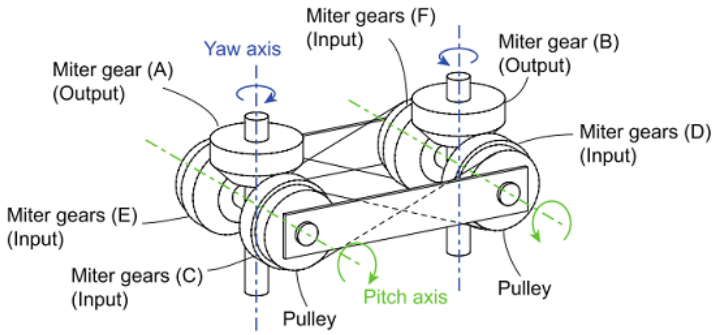


Figure 2.17: The proposed new 2-DOF joint mechanism (retrieved from [58]).

Experiments were conducted on the robot to test the rotation power of the modular design and to perform some snake motions. A cable was used to power the robot and the movements generated by the sidewinding gait were generated from the computer to the robot. This experiment concluded that the modular design is capable of performing a large range of motions, is suitable for miniaturisation, gives torque power equivalent to two motors for rotation in pitch and yaw axes, and a flexible tube can be easily used to cover the robot from dirt [58].

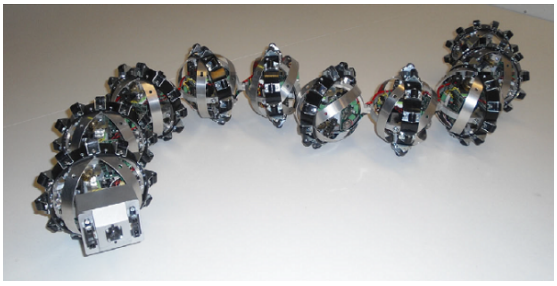
Figure 2.18: The miniature ACM snake robot (retrieved from [58]).

2.4.3 NTNU/SINTEF Snake Robots

Most of the snake robots developed at NTNU/SINTEF were designed with 2-DOF rotation joints. Figure 2.19 presents the four snake robots developed at NTNU/SINTEF. The first snake robot developed at the institutes was a water hydraulic snake robot called Anna Konda [69]. The robot was designed to operate during fire events for search-and-rescue, and firefighting operations. The head module of the robot was equipped with two nozzles which could spray water supplied from a pressure hose via a dedicated valve. The next snake robot, called Aiko, was developed to be more portable using DC motors as the actuators [40].

(a) Anna Konda (retrieved from [69])

(b) Aiko (retrieved from [40])



(c) Wheeko (retrieved from [79])

(d) Kulko (retrieved from [79])

Figure 2.19: Snake robots used as research platforms at NTNU/SINTEF.

A new robot, Wheeko, was then developed to study slithering locomotion [79]. Slithering locomotion was achieved by the presence of passive wheels which provided anisotropic friction. Though the interior modular designs of Wheeko and the robot developed next, Kulko, were similar [79], the module of Wheeko was equipped with small passive wheels whereas the module of Kulko was installed with Force-Sensitive Resistor (FSR) sensors that enabled the module to sense the magnitude and direction of the forces acting on it.

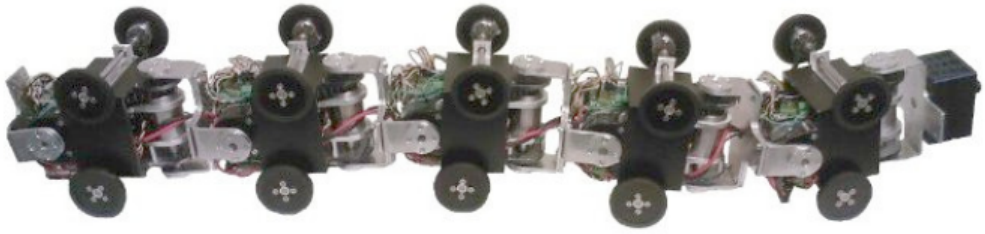


Figure 2.20: NTNU/SINTEF snake robot, PIKo (retrieved from [80]).

Mamba is the latest snake robot developed at the institutes to support ongoing research on snake robot locomotion, extending to underwater locomotion. The robot has two main features. First, the robot is waterproof with the ability to submerge at least 2 m underwater. Second, the force and torque exerted on each module of the robot can be measured relatively accurately with strain gauges, to perform adaptive locomotion [70]. Each module has a hobby servos connected to the aluminium casing equipped with strain gauges. Besides the strain gauges, each module is also equipped with two temperature sensors, a 3-axis accelerometer and a water leakage detector. The tail module only contains a mechanism holding the power cord, while the head module contains a microcontroller card, a wireless camera, inertial sensors, a Wi-Fi module for communication with external computer, and four RGB LEDs for camera lighting and to indicate status. The motor in the module is controlled by a microcontroller which communicates with other modules via a CAN bus. The waterproof feature consists of a rubber seal to seal the end of the joint shaft, O-ring to seal the inspection lid and a rubber plug to seal the entry point of power and the CAN bus. The Mamba snake is designed to have five configurations as shown in Figure 2.22.

Figure 2.21: NTNU/SINTEF snake robot, Mamba (retrieved from [70]).

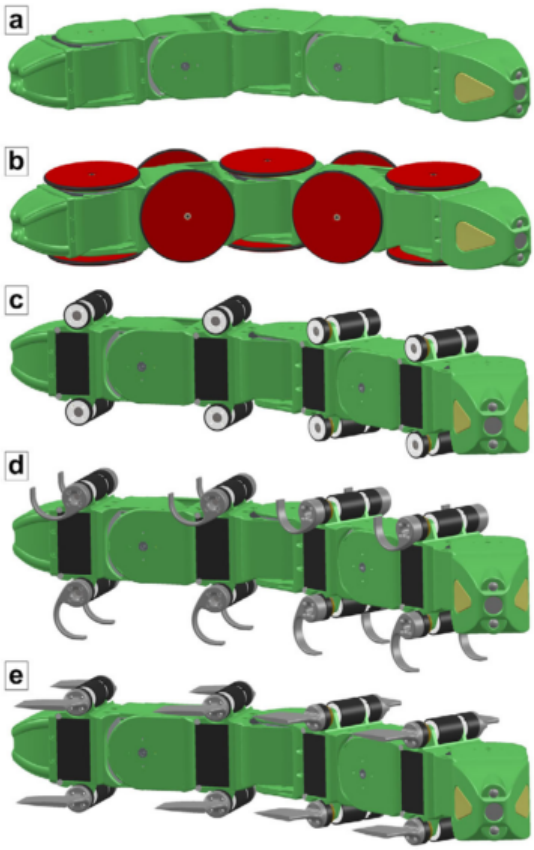


Figure 2.22: Configurations of the Mamba Snake Robot: a) Normal, b) Passive disc wheels, c) Active wheels, d) Motorised legs, e) Motorised Fins (retrieved from [70]).

2.5 Summary

Snakes have joints which can rotate horizontally and vertically. Snakes utilise gaits such as lateral undulation, concertina, rectilinear and sidewinding. These gaits are efficiently generated with the help of the snakes' scales and muscles. The scales provide anisotropic friction between a snake's belly and the ground whereas muscles can control the scales and generate stretching and pulling motions in the snake's body.

Nevertheless, snake robots do not have scales and muscles. With the absence of scales and muscles, snake robots require passive wheels to perform lateral undulation gait. Snake robots are capable of performing linear progression gait, which resembles concertina and rectilinear gaits, but are more efficient and do not require muscles. Snake robots also have the capability of performing gaits that cannot be performed by biological snakes; these gaits are rotating and rolling gaits.

Lateral undulation and linear progression gaits are considered 2D locomotion and the others are considered 3D locomotion. There have been extensive studies conducted on lateral undulation gait to model the motions. As compared to the serpenoid curve, the serpentine curve model describes the motions of lateral undulation gait closest to ones performed by biological snakes. Most of the models are based on the curve models and assume that snake robots are applied with side-slip constraints. Some of the models also consider lateral undulation with sinus-lifting. On the other hand, there are a limited number of studies conducted on modelling for other gaits because, during 3D locomotion, robots' joints are not constantly interacting with the ground. A simplified kinematic motion model, which is based on the virtual chassis [33], is capable of modelling rolling, slithering, sidewinding and rotating gaits with reasonably accurate results [34]. A velocity model, which is based on FFR [35], was used to model the velocity of sidewinding gait with good accuracy but without describing the travelled direction and orientation of the robot. The accuracy of the model decreases as the speed of the gait increases and is no longer valid when the wavelength of the gait decreases.

The locomotion of snake robots can be generated either by closed-loop or open-loop control. One common form of closed-loop control is the model-based control that is common for lateral undulation gait. Another form of closed-loop control is the obstacle-aided control that efficiently utilises obstacles to propel a snake robot forward. With the obstacle-aided control, a snake robot needs to detect its surroundings. This can be

achieved by utilising tactile sensors or compliant behaviour in the robot's joints. CPG control is a form of open-loop control that generates cyclic movement where its shape and speed can be altered. A sinusoidal function generator control is very similar to CPG-based control with simpler parameter control. This control approach is commonly used to generate linear progression, turning, sidewinding, rotating and rolling gaits, which are generated by setting the parameters to certain values. An approach that utilises the virtual chassis was proposed to direct the robot to follow a trajectory using various gaits, which are used to control the motion of the robot.

Snake robots can be divided into five categories: robots with passive wheels, robots with active wheels, robots with active treads, robots with 3D joints and robots with linear expansion joints. Ideally, a snake robot should be small, should move moves fast, should be capable of performing various gaits, and should have a long operating time. The requirements of fulfilling the criteria depend on the application of the robot, and setting the requirements high can significantly increase the development cost of the robot. When designing the module, it is important to consider the joint type, the radius and the length of the module. When designing a snake robot to operate for a specific application, it is important to determine the type of power source, communication, and sensor and instrumentation suitable for the application.

The snake robots developed by other research institutes have a variety of designs and have demonstrated their specialisation for various applications. CMU snake robots consist of pitch and yaw modules with 1-DOF rotation joints that can rotate $\pm 90°$. This large angle of rotation enables the robots to perform gaits such as pole climbing. The series elastic joint was developed to estimate the torque exerted to the robot's joints. The ACM snake robots are equipped with passive wheels in order to perform lateral undulation gait. One of the robots, which is equipped with fins, is capable of swimming under water. Most of the robots have 2-DOF rotation joints, and this limits the range of angle the joints can rotate and increases the module size of the robots. A 2-DOF joint mechanism was proposed to minimise the length module of a snake robot. Some of the NTNU/SINTEF snake robots are similar to others, but there are a few snake robots with differences worth mentioning. NTNU/SINTEF developed a hydraulic actuated robot that is capable of spraying water to extinguish a fire. A specialised snake robot, consisting of 2-DOF rotation joints and active wheels, was also developed to navigate in complex piping systems. NTNU/SINTEF also developed a waterproof snake robot that is capable of sensing the force exerted to its joints using strain gauges.

Chapter 3

Design of a Modular Snake Robot

3.1 Introduction

A robust snake robot with a reliable control software was designed and developed as the robotic platform for this thesis. Using a robot simulator would not be adequate to validate the motion analysis because the simulators are not capable of simulating complex dynamic effects which exist in the real world.

Based on the previous chapter, the design of the snake robot can be divided into categories depending on the application of it. This thesis focuses on a snake robot for search-and-rescue operations and structural inspections so the snake robot needs to possess multiple gait capabilities. Using passive wheels underneath the robot will be risky as it increases the odds for the robot to get stuck to obstacles during the operations. The robot designed for this thesis was highly inspired by the modular snake robots developed at the CMU [57, 53]. The modular snake robots were made up of several modules connected in pitch-and-yaw order. In other words, every consecutive module rotates in the axis 90° out of phase to one another.

This chapter covers the development of mechanical parts, electrical and electronics, and software of the developed modular snake robot. This first section reviews the design process and the progress of the modular prototypes achieved throughout the research. The mechanical designs of the modular snake robots and the results of the Finite Element Analysis (FEA) of robots' joints are presented. The manufacturing of the mechanical parts with a low-cost 3D printer is also discussed. In the second section, the electronics and actuators used for the snakes are reviewed. As for the final section, the software architecture of the snakes which controls their locomotion is presented.

3.2 Design Process

The design and development of the modular snake robot for this thesis was conducted according to the engineering design process proposed by Ertas and Jones [81] but with a slight modification to fit it to this research. The engineering design process was divided into several stages: problem statement, research, concept design, preliminary design, detailed design, prototype manufacturing, and testing and design evaluation, as shown in Figure 3.1. This design process was highly iterative for the mechanical, electrical and electronics, and software aspects of the snake robot. Throughout the thesis, four design prototypes were developed, as shown in Figure 3.2. The use of a low-cost 3D printer had significantly reduced the cost of manufacturing the mechanical parts of the prototype modules.

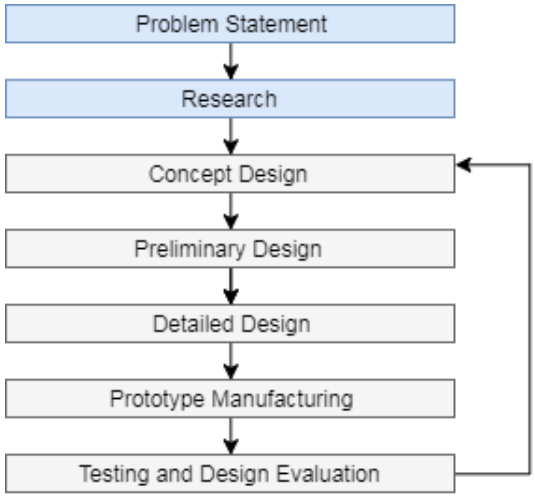


Figure 3.1: The modified design process of the modular snake robot.

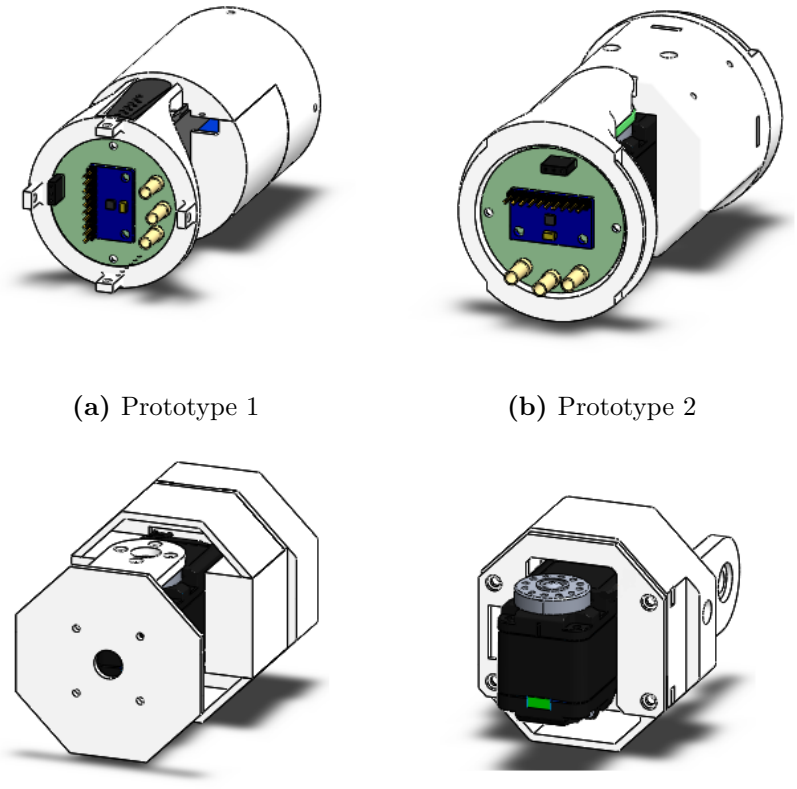
The first modular prototype used a low-cost hobby servo, HXT12K. The servo used in the design was unreliable because each servo had inconsistent maximum holding torque. The module was also quite heavy as the design only focused on accommodating the spaces for electronics.

The second modular prototype used a smart servo, Herkulex DRS-0101, instead of the hobby servo similar to the snake robot designed by Melo et al. [82]. This was because the price of the smart servo was reasonably close to the price of a mid-range hobby servo. In addition, the smart servo has inbuilt feedback so there was no requirement to include additional sensors to the design. However, the weight and the length of the module was still an issue with the design. The weight and length of the module required a significant amount of torque for the servo to rotate the joint. The designs of the first two prototypes had several critical issues when they were tested individually, and they were not used to

build the snake robots.

The third modular prototype managed to significantly reduce the weight of the module. The weight was reduced by breaking down the modular design into several more parts and each part was designed with a single-layered-wall. This modular design prototype was used to build the first prototype modular snake robot. At one point of the research, it was decided that the batteries should not be included in the modular design as the battery compartments were taking up most of the space in the module. Thus, the modular snake robot was tethered by a cable which provided power and communication. Furthermore, the head module was designed and manufactured so that the actuator of the second module could not be utilised for locomotion.

The fourth modular prototype was designed and manufactured to resolve some issues which were not addressed in the previous prototypes. The second prototype modular snake robot, which has been used throughout this thesis, was built using this modular prototype. Further details of the design are discussed in the next section.



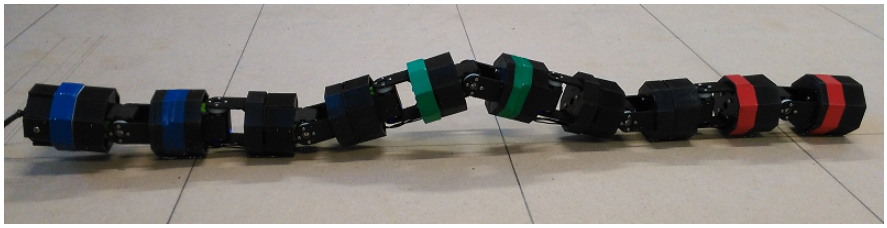
(a) Prototype 1 **(b)** Prototype 2

(c) Prototype 3 **(d)** Prototype 4

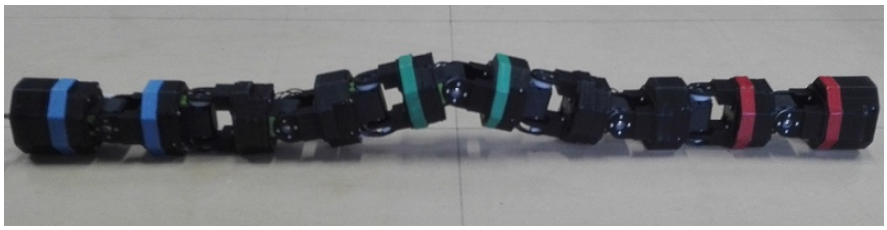
Figure 3.2: The progression of the modular prototype from (a) to (d).

3.3 Mechanical Design

The last two modular prototypes were assembled into pitch-and-yaw modular snake robots, as shown in Figure 3.3. The first and the second prototype robots were assembled using the third and the fourth prototypes respectively. A head (first) and a tail (last) modules were also designed and manufactured and they were assembled at each end of the modular assembly. The design specification of the robots are presented in Table 3.1. The next subsections present and discuss the design of the modules used to assemble the second prototype robot.



(a) The first prototype modular snake robot



(b) The second prototype modular snake robot

Figure 3.3: The progression of the prototype modular snake robot from (a) to (b).

Table 3.1: The design specification of the prototype modular snake robots.

Specifications	First Prototype	Second Prototype
No. of modules	10	
Diameter (mm)	56	
Length (mm)	800	680
Mass (kg)	0.9	
Max holding torque (Nm)	1.2	
No load speed (rpm)	60	
Max voltage (V)	12	
Max current (A)	10	
Communication	TTL	TTL and RS485

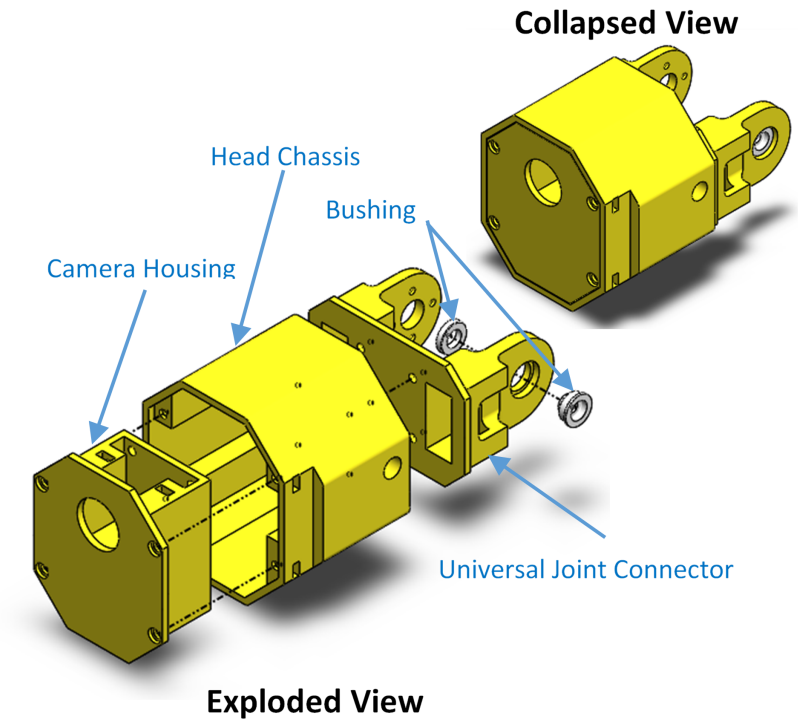
3.3.1 Head and Tail Modules

The head and the tail modules are necessary for the robot to perform efficient locomotion. Without the head and the tail modules, the actuator of the first pitch will not be utilised and the length of the last pitch module will not be equal to the rest of the modules. The designing and manufacturing of the head and the tail modules only started after the testing and validation of the third modular prototype. Therefore, there are only two design iterations for the head and the tail modules. The only major difference between the first and the second prototypes is that the parts are easier to assemble into a module with the latter design, as shown in Figure 3.4.

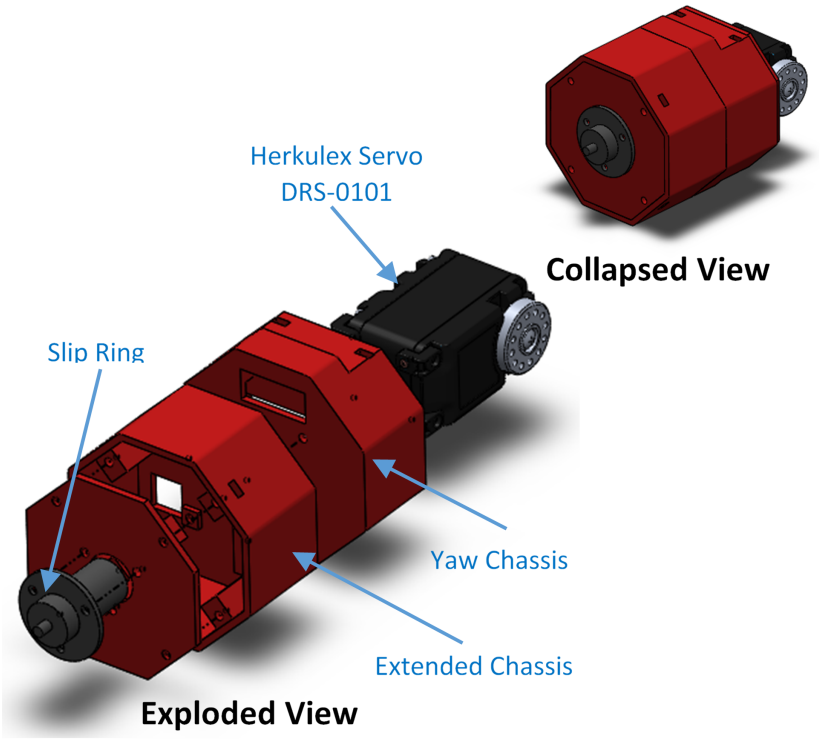
The first module of the robot, also known as the head module, is a passive module. It has no actuation, and was designed specifically to accommodate a camera, similar to the head modules of the CMU, the Mamba, and the ACM snake robots [64, 70, 83]. The head module is lighter than the rest of the modules, weighing 68.1 g because there are not many other components in the module besides a microcontroller and a camera.

The last module of the robot, also known as the tail module, has a similar design as the rest of the modules but with an extended part designed to hold a slip ring and some electronics. This module is the first module that receives power from the power supply and communication from the computer. The slip ring allows wires to rotate preventing them from getting twisted while the snake robot performs a rolling gait. Based on a test conducted in the lab, the use of the slip ring showed no adverse effects on the communication and power of the snake. The tail module is slightly heavier than the rest of the module weighing 108 g.

The difference in the mass of the head module and the rest of the modules could easily be controlled by adding more weights in some spaces available to it. However, it would be hard to control the mass of the tail module as it is the heaviest. The easiest solution would be to add more weights in the rest of the module to make their masses close to that of the tail module. However, this would require redesigning the rest of the module to be larger so that there would be more spaces to add weights to them. Though this difference might affect the locomotion of the snake, it would be an interesting effect to investigate in the following chapters.



(a) The assembly of the head module



(b) The assembly of the tail module

Figure 3.4: The assemblies of the head and the tail modules.

3.3.2 Pitch and Yaw Modules

The joint connector in the third prototype had broken several times due to the impacts that occurred during the locomotion. In the fourth prototype, the issue was resolved by increasing the thickness of the joint design. The strength of the joint for both prototypes are analysed and compared in the later section. Since the battery compartments in the third prototype were of no use, the fourth prototype discarded the compartments and fully utilises the remaining space for electronics. Furthermore, the length of the module was shortened from 80 mm to 68 mm, which increases the manoeuvrability of the snake robot in confined spaces.

The difficulty of assembling the module parts together and from one module to another was resolved in the fourth prototype by embedding nuts into the 3D printed parts, as shown in Figure 3.5. The difficulty in the inter-module connection was resolved by moving the joint of the module to the back instead of being at the servo, as shown in Figure 3.2d. Unlike in the previous prototypes, the pitch and the yaw modules were designed slightly differently to allow neat cable management, as shown in Figure 3.6. The pitch and the yaw modules are designed 90° out of phase so that the cable, which connects between one module to another module, is maintained at the bottom of the main frame and the joint of each module, as illustrated in Figure 3.5.

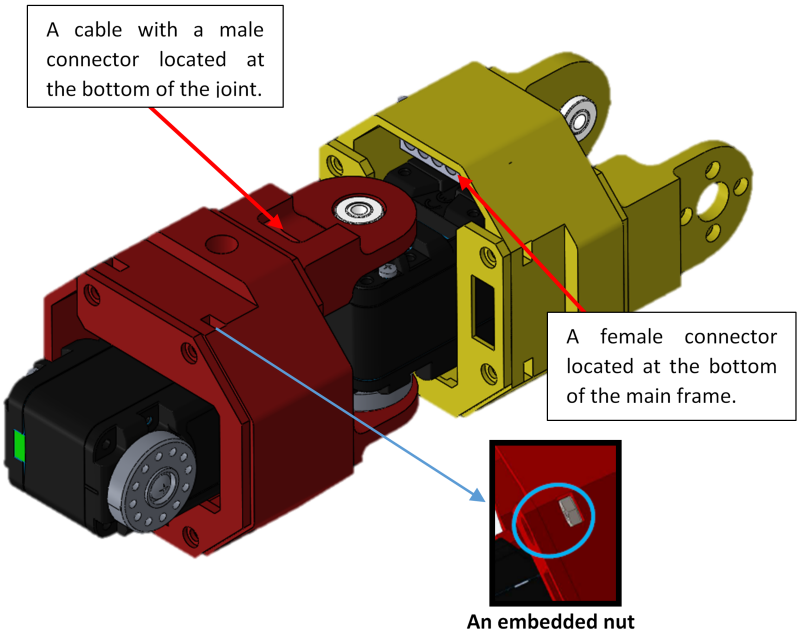
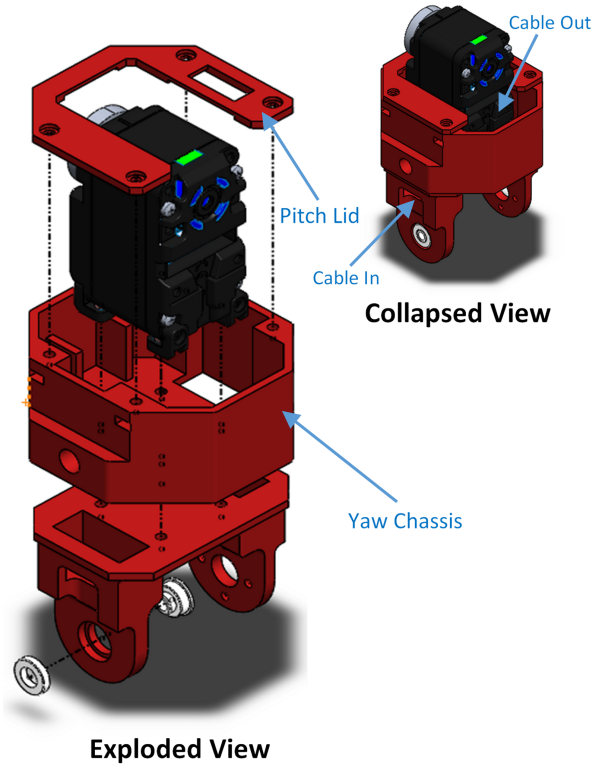
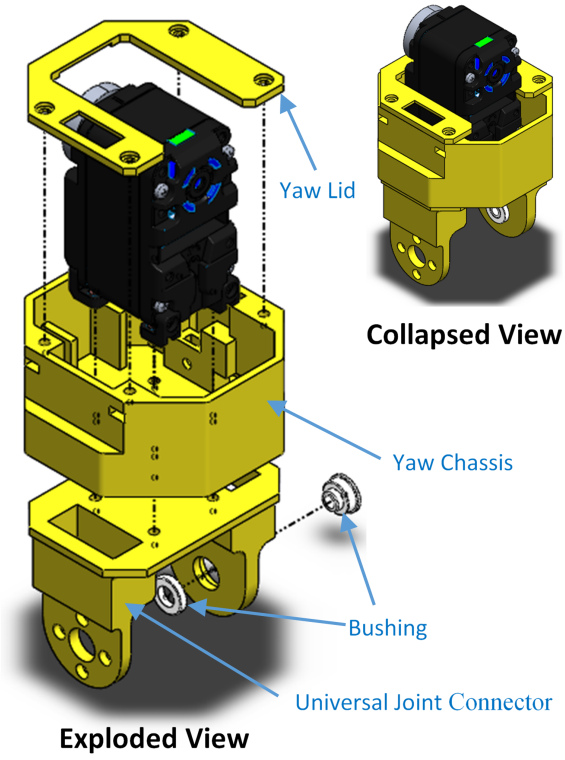


Figure 3.5: The cable connection is maintained at the bottom of the main frame and the joint of each module.



(a) The assembly of the pitch module.



(b) The assembly of the yaw module.

Figure 3.6: The two modules are designed to be 90° rotated with respect to one another.

3.3.3 3D Printer Manufacturing

The mechanical parts of the prototype modular snake robots were manufactured using a low-cost 3D printer, D-Force (V2, D-FORCE.TW), as shown in Figure 3.7. The size of the extruder of the 3D printer was 0.4 mm, and the 3D printer required 1.75 mm filament. Though the 3D printer printed the part at low-cost, it could not print support material. Without support material, the mechanical parts were printed in a way that the material would not collapse while it was soft. Polylactic Acid (PLA) was used as the 3D printing material as it is more environmentally friendly compared to Acrylonitrile Butadiene Styrene (ABS) [84].

Figure 3.7: The low-cost 3D printer, D-Force, used to manufacture the mechanical parts.

To print the mechanical parts, the 3D models of the parts designed with Solidworks were converted to STL format. An STL file describes the mesh of an object in terms of 3D coordinates of vertices which form the polygonal surfaces of the mesh. The information in the STL file alone would not be sufficient to be used for 3D printing. Using a slicing software, the information in the STL file was converted to a set of instructions which were written in G-Code script. Since coordinates in the STL file were dimensionless, it was important to ensure that the Solidworks was set to a similar measurement scale.

The slicing software was responsible for coordinating the movement of the 3D printer and setting the temperatures of the extruder and the heated bed of the 3D printer along

with many other variables which may affect the printing quality [84]. The slicing software also enabled the object to be printed in various orientations and locations of the printing platform. Based on Figure 3.8, when printing without support material, the orientation of the yellow object was more sensible than that of the pink object. It is also important to be aware that different slicing software may result in various 3D printing results as found in [84]. The slicing software used to manufacture all the parts of the snake robots was Slic3r, and the printing results for all parts were reasonably good.

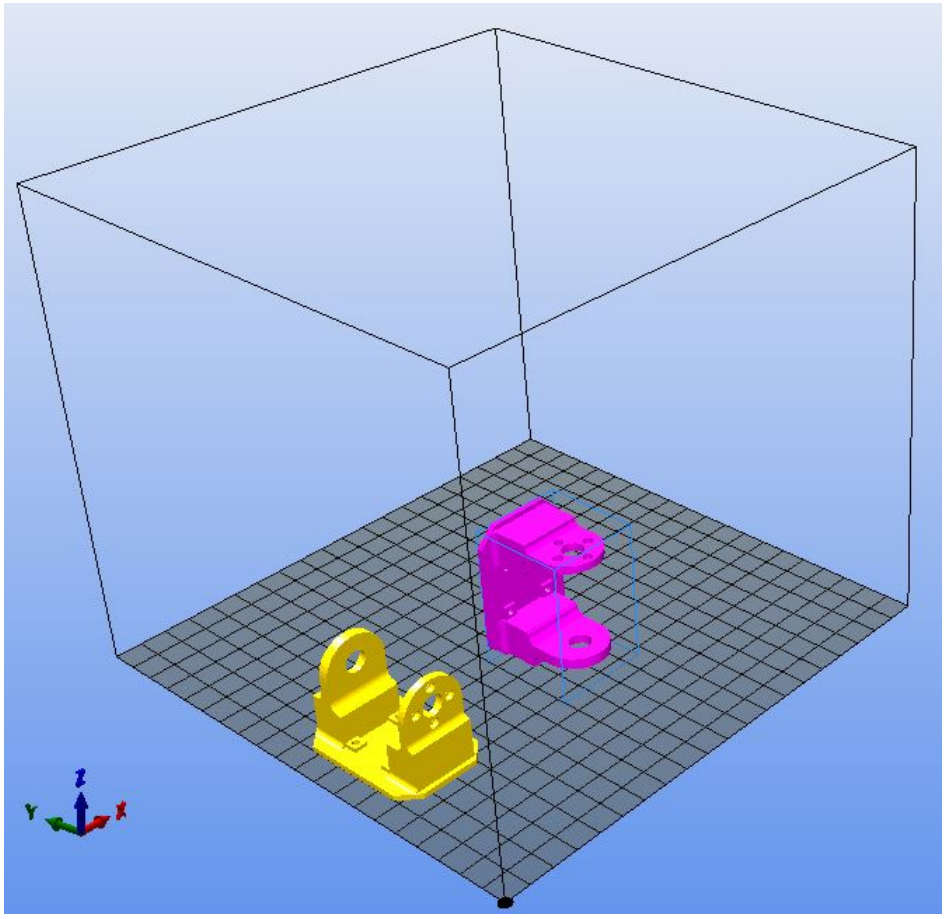


Figure 3.8: Two same objects on the printing platform at different orientations as illustrated in the slicing program.

3.3.4 Joint Strength

Finite Element Analysis (FEA) was performed on the joints of the snake robots using SolidWorks. There were two types of effects considered in the analysis: bending and torsion. Bending occurs when the force exerted is parallel to the rotation axis of the joint, whereas torsion results from the torque exerted to the joint. The assumption made for this analysis was that the maximum torque exerted at the joint would be 1.2 Nm (the maximum torque of the Herkulex Servo DRS-0101) for the worst case in both effects. In bending, the coupling force reacting at the top and the bottom of the joint was 31.6 N. Table 3.2 and Table 3.3 respectively present the results of the stress (von Mises) and displacement (URES) that occurred at the joint due to torsion and bending.

Based on the study conducted in [85], the ultimate tensile strength of the PLA filament used as the 3D printed material depends on the orientation angles which the stress is acting relative to the grain direction of the 3D-printed parts. Due to the limited capability of the low-cost 3D-printer used to print the first and second prototype snake robots, the joints were printed with an orientation angle of 90°, as shown in Figure 3.9.

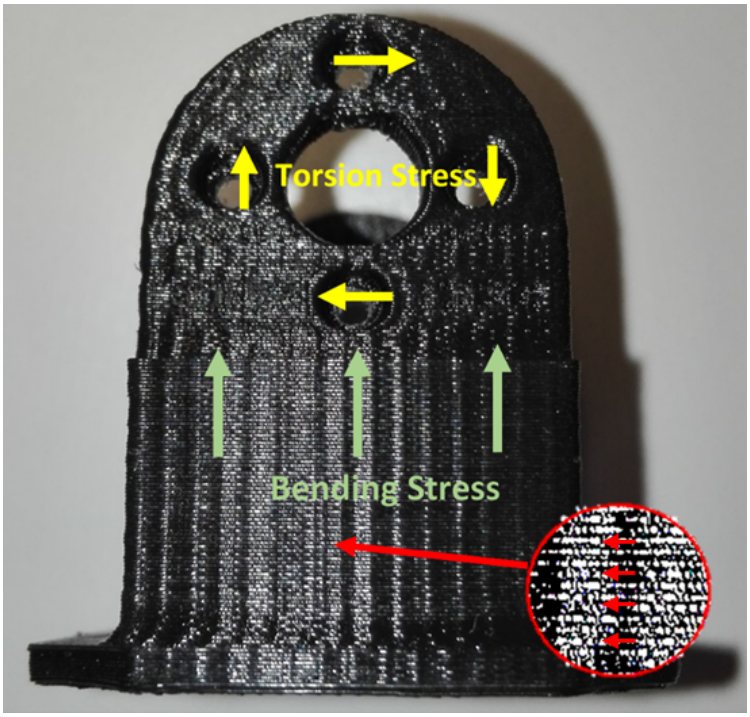


Figure 3.9: In both bending and torsion, most stress would be acting perpendicular to the direction of the grain.

According to Letcher and Waytashek [85], an object printed at this orientation angle would have weaker strength. They found that the ultimate tensile strength and the ul-

timate bending stress for 90° were 54 MPa (approx. 8.5% smaller than the general PLA mentioned in [86]) and 86 MPa [85]. Since the deformation due to bending and torsion could occur simultaneously at the joints of the snake robots, it would be safe to use the ultimate tensile strength to determine the maximum stress allowable in the analysis. Using a safety factor of 3, the maximum stress allowable would be 18 MPa respectively. The safety factor was set high to consider the yield strength of the material and the quality of the 3D printer since the 3D printer used to print the joints had a lower quality than that used by Letcher and Waytashek [85].

Based on the results presented in Table 3.2 and Table 3.3, the stress from both torsion and bending in the third and the fourth modular prototypes was below the maximum allowable stress. The stress in the third modular prototype exceeded the maximum allowable stress which explains the failures that occurred in the joints.

Table 3.2: The stress and displacement due to torsion factor in the fourth modular prototype were smaller than that of the third modular prototype.

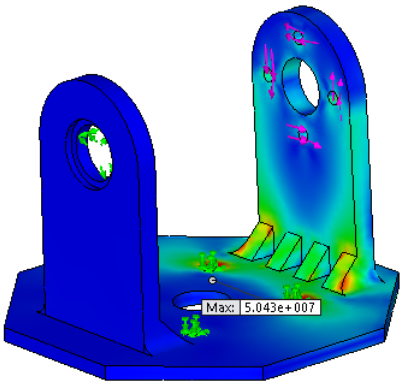
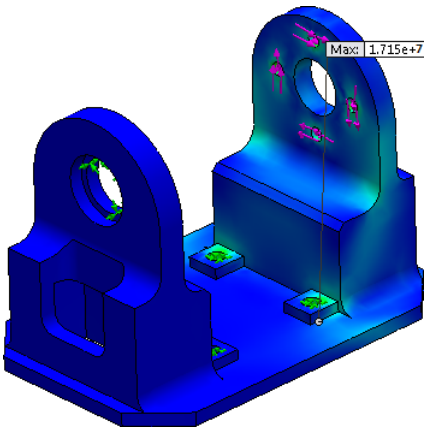
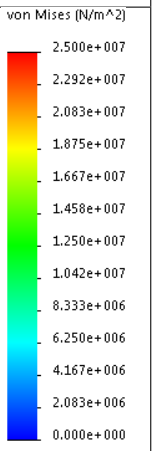
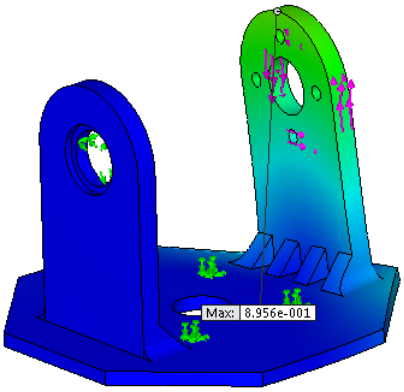
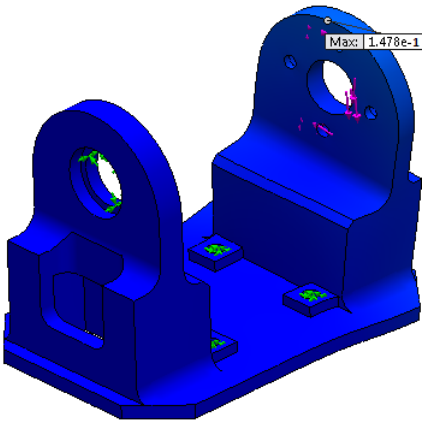
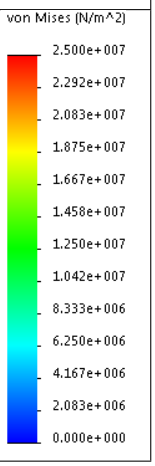
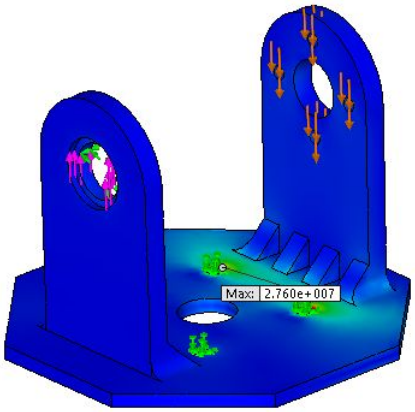
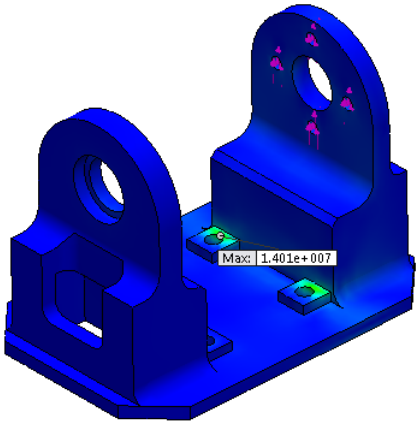
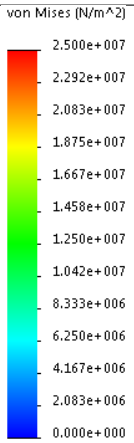
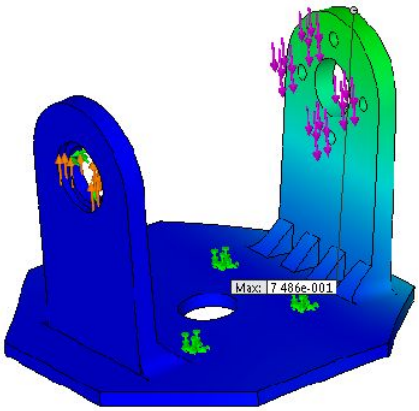
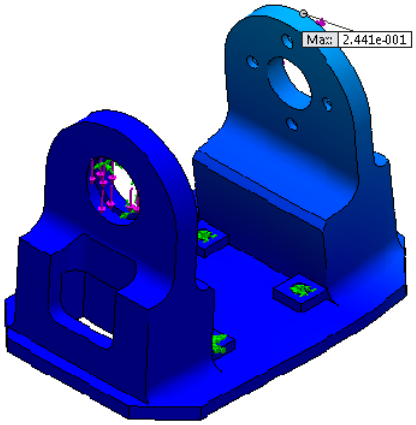
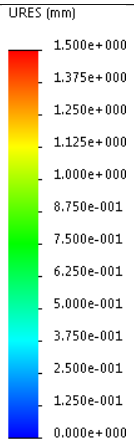
Prototype 3	Prototype 4	Colour Bar
		
		

Table 3.3: The stress and displacement due to bending factor in the fourth modular prototype were smaller than that of the third modular prototype.

Prototype 3	Prototype 4	Colour Bar
		
		

3.4 Electronics

3.4.1 Actuator

The actuator used in each module of the snake robot was the Herkulex smart servo motor, which can be connected to two other smart servos to form a parallel bus connection. This parallel bus connection has a pair of wires for communication and another pair of wires for power, as shown in Figure 3.10.

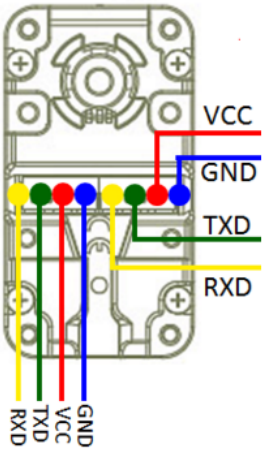


Figure 3.10: A schematic diagram showing the wire interface of Herkulex smart servo.

Table 3.4 shows the specification of the smart servo. Since the smart servo is a DC motor, the ideal DC motor characteristics could be used to estimate the torque and rotation speed at the maximum power [87]. Based on Figure 3.11, the torque and rotation speed would respectively be 0.6 Nm and 30 rpm at the maximum power.

Table 3.4: The specification of the smart servo motor, Herkulex DRS-0101.

Specifications	Herkulex DRS-0101
Voltage (V)	7.4 (nominal)
Stall Torque (Nm)	1.2
No load speed (rpm)	60
Resolution (°/bit)	0.325
Operating Angle(°)	±160

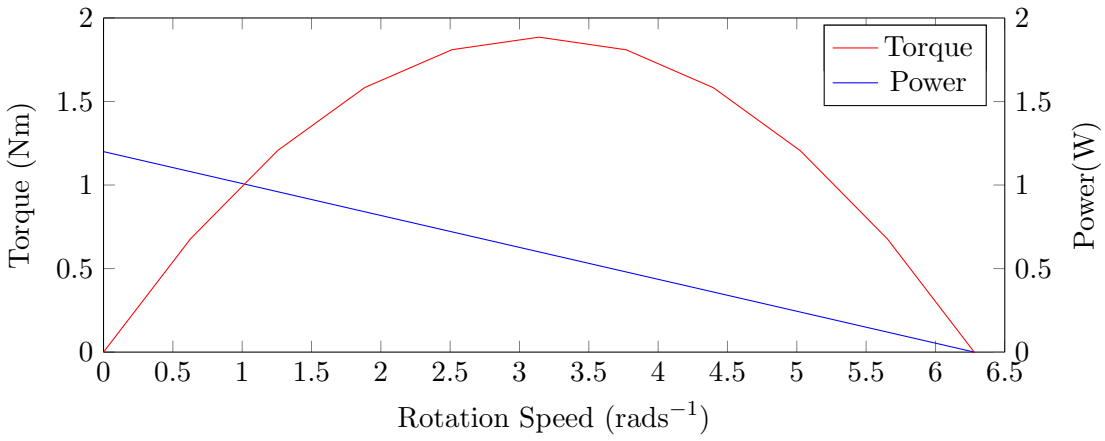


Figure 3.11: Theoretical Herkulex DRS-0101 Servo Performance.

The capability of the servo to lift the number of modules can be determined by assuming that the snake robot is like a cantilever beam, as shown in Figure 3.12.

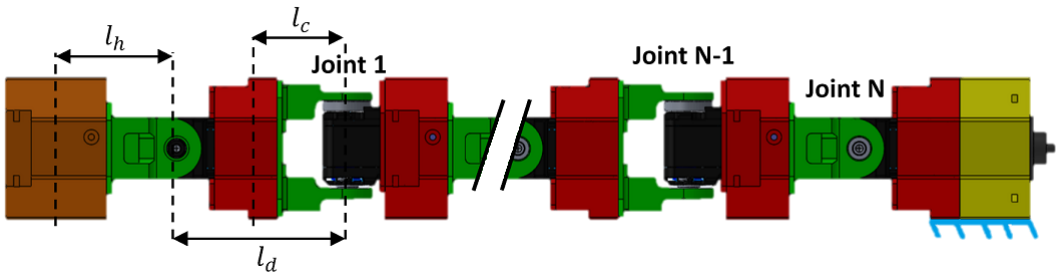


Figure 3.12: The body diagram of the snake robot acting similarly to a cantilever beam.

The torque of the servo supporting the suspended modules can be calculated using Equation 3.1 where the notations are defined in Table 3.5.

$$T_{hold} = m_h g\big(l_h + N l_d\big) + m_c g\big((N-1)! l_d + N l_c\big) \tag{3.1}$$

T_{hold} for N ranging from 1 to 10 were iterated using the mechanical properties of the modular snake robot as presented in Table 3.5, and the results are shown in Figure 3.13. Based on Figure 3.13, the maximum number of the modules (including the head module) that the Herkulex servo motor can hold and drive are four and three respectively. The capability of the actual snake robot was tested, and it showed a similar outcome.

Table 3.5: The mechanical properties of the modules of the snake robot

Notation	Definition	Value	Unit
m_h	The mass of the head module	0.0681	kg
m_c	The mass of the pitch/yaw module	0.09	kg
l_d	The distance between two consecutive modules	0.0681	m
l_c	The distance between the shaft of the pitch/yaw module and its centre of mass	0.02403	m
l_h	The distance between the shaft of the head module and its centre of mass	0.02404	m
g	The gravitational acceleration	9.81	m/s^2

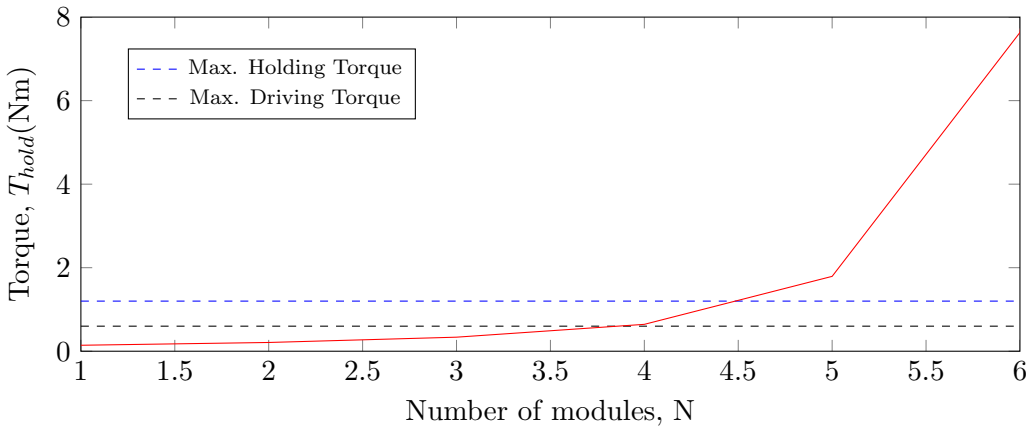


Figure 3.13: The torque required to hold N number of modules including the head module.

3.4.2 Communication

The smart servo is capable of reading the angle and power feedback. The electronic design for the first prototype snake robot used the parallel bus connection available, as shown in Figure 3.14. The Arduino Atmega 2560 communicated with the servo motors using TTL serial method. There were several issues with using this communication method that affected the locomotion of the robot. Disruption in the communication, which caused the motion of the robot to become unreliable, occasionally occurred when the wire was disturbed during the operation. Another issue with the communication method was that the maximum communication speed of 0.66M baud was not fast enough to support more than ten servo motors in transmitting and receiving data at more than 10 Hz. There was also an issue with the power feedback. It could saturate easily while the motor was performing motion generated by a sinusoid generator; this made the smart servo unreliable to detect obstacles. Thus, these issues limited the snake robots to transmit their joint angles at 10 Hz only.

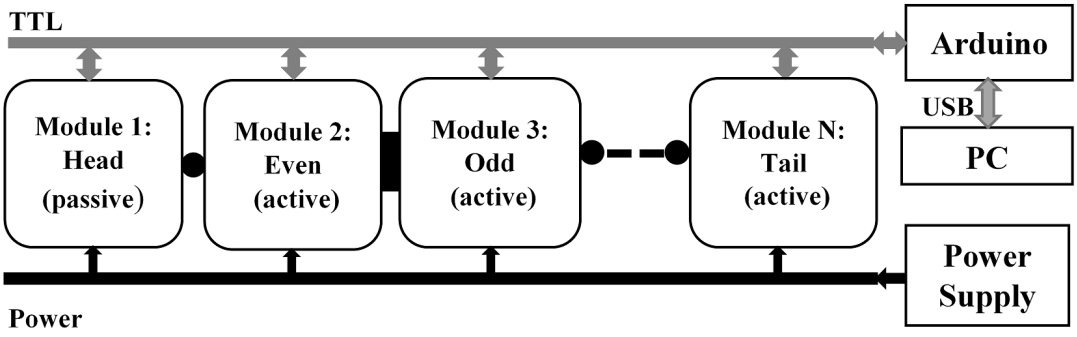


Figure 3.14: The hardware architecture of the first prototype snake robot.

In the second prototype snake robot design, the hardware architecture had three electronic parts added to each module as shown in Figure 3.15: a microcontroller (Teensy LC), RS485 board and an inertial measurement unit (IMU). The microcontroller reads the data from the IMU and transmits it to the main computer via RS485 communication. The RS485 board plays an important part in converting the UART signal from the microcontroller to RS485 signal. The advantage of using RS485 over I^2C is that it can transmit signals more reliably over longer distances. Adding an IMU was crucial for estimating the orientation of the snake robot for future research.

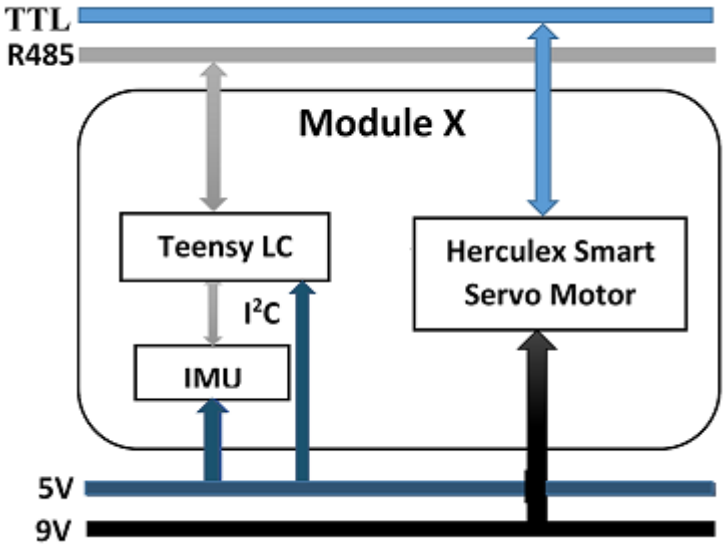


Figure 3.15: Modularisation of electronics in the module of the second prototype snake robot.

3.5 Software

Robot Operating System (ROS) is the robotic middleware that was used to develop the robot application that operated the first and the second prototype snake robots. ROS is "a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms" [88]. 'Nodes' are processes that perform computation and communicate with other 'nodes' through 'topics'. They communicate with each other by passing messages. A message is simply a data structure, comprising typed fields. A 'node' can publish a 'topic' as well as subscribe to a 'topic' to receive messages. The 'topic' is used to identify the content of the message. All the connections are managed by the 'ROS Master' which enables each node to locate one another. It provides naming and registration services to the rest of the nodes in the ROS system. The interactions

between the nodes can be summarised, as shown in Figure 3.16. When a 'node' is activated, the 'node' declares the 'topic' which it will be subscribing to and advertising to. Then a communication between two 'nodes' is established while the 'nodes' are enabled. This system allows the Graphical User Interface (GUI) to run in parallel with the joint controller of the snake robots.

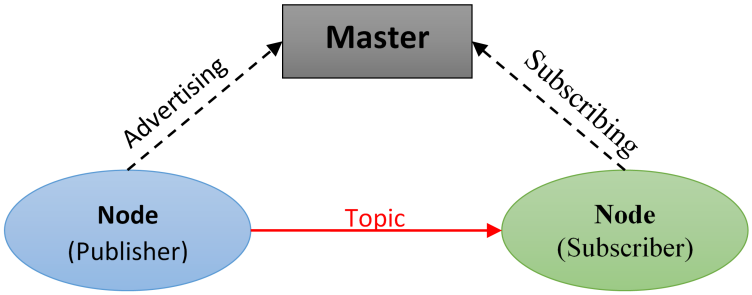


Figure 3.16: The interactions between Node-Node and Node-Master in ROS.

Figure 3.17 shows the nodes and topics in the communication network of the snake robot control system. "/my_plugin" is the node running a GUI which enables an operator to set inputs from the GUI and sends the information to "/gait_generation" and "/serial_node" through some topics. "/gait_generation" mainly performs calculation based on the inputs it receives from "/my_plugin" and the outcome of the calculation is sent to "/serial_node". "/serial_node" receives results from "/gait_generation". "/serial_node" also receives the motor speed setting from the GUI directly. The major role of "/serial_node" is to bridge the communication between the computer and the Arduino. The Arduino acts as a bridge between the serial servo motors and the ROS by performing low-level tasks such as controlling the joint angles and receiving feedback joint angles from the robots.

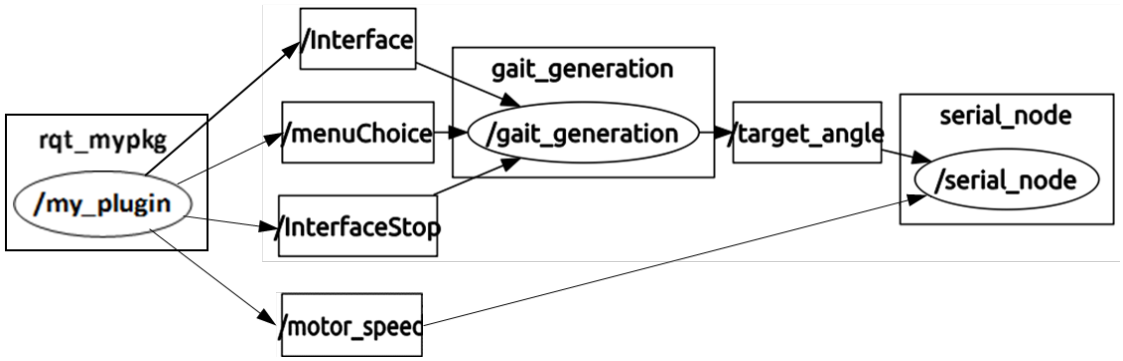


Figure 3.17: The communication network of the snake robot control system in ROS.

A GUI was designed and developed to create a friendly and intuitive method for

controlling the locomotion of a modular snake. The GUI was developed using 'rqt', a "Qt-based framework for GUI development for ROS" [89]. The following are the specifications of the GUI:

1. Stop, pause, and reset buttons.
2. Selection menu to choose the gait modes such as linear-progression, rolling and rotating/side-winding.
3. Slider to change the motor speed.
4. Arrow keys to change the direction of the locomotion.
5. Display showing the angular feedback from the servo motors.

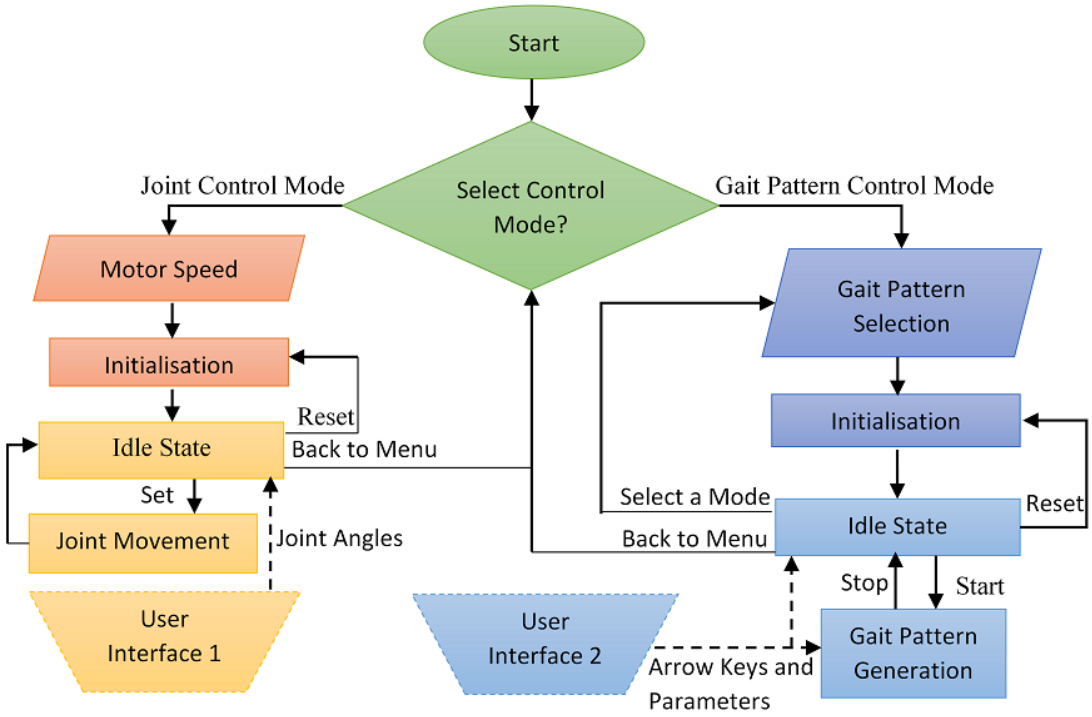


Figure 3.18: The flow chart of the modular snake robot control interface.

The flow chart of the GUI is shown in Figure 3.18. The GUI provides two control modes: Joint Control and Gait Pattern Control. When the program starts, a user needs to choose either mode, as shown in Figure 3.19.

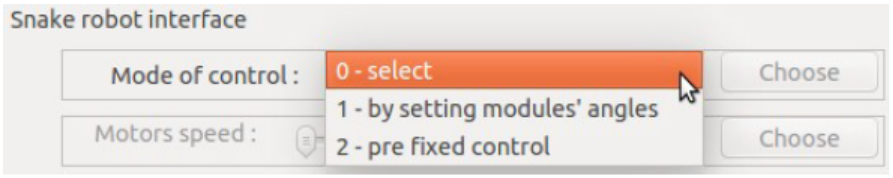


Figure 3.19: Control modes of the modular snake robot (1 - Joint Control and 2 - Gait Pattern Control).

In Joint Control mode, the user has to set the motor speed to the desired speed before the "Initialisation" state. The speed can be set using the slider, as shown in Figure 3.20. As for Gait Pattern Control mode, the user has the options to choose a gait. During the "Initialisation" state, the robot's joint angles are set to zero. The robot will then enter into the "Idle" state. In this state, the user can press "Initialisation" (Reset) button to begin the "Initialisation" state or press "Back to Menu" button to choose the control mode.

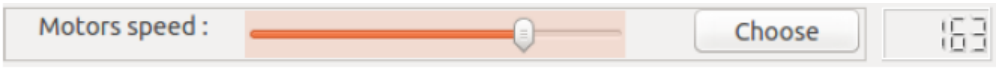


Figure 3.20: Speed setting of each servo in the robot's joints.

During Gait Pattern Control mode, the user can also press the "Select a Mode" button while in the "Idle" state to change the gait of the robot, as shown in Figure 3.21. During Joint Control mode, the robot moves when the button, "Set", is pressed. Each joint stops moving after it reaches to the joint angle set in User Interface 1. During Gait Pattern Control mode, the robot moves continuously according to the joint angles generated by the gait when the "Start" button is pressed and enters the "Idle" state again when the "Stop" button is pressed. Unlike in Joint Control mode, User Interface 2 continuously feeds inputs to the robot while in the "Idle" and the "Gait Pattern Generation" states, meaning that the shape of the robot can still be changed according to the gait parameter inputs and the control keys. This feature allows the user to change the gaits and movement direction of the robot without restarting the program.

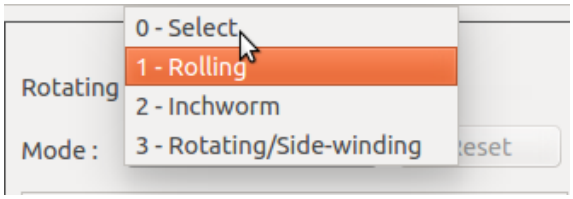


Figure 3.21: The interface to select the gait mode.

The feedback joint angles of the robot and the buttons "Reset", "Set", "Stop" and "Back to menu" are as shown in Figure 3.22. The control interface for each gait is presented in Figure 3.23. There are two disadvantages of the GUI. One of them is that the gait of the robot cannot be switched to another gait without going back to the "Initialisation" state again. Another disadvantage is that the robot does not make smooth transition when the gait parameters are changed during the locomotion. This none-smooth transition also causes the robot to straighten its body abruptly during the "Initialisation" state.

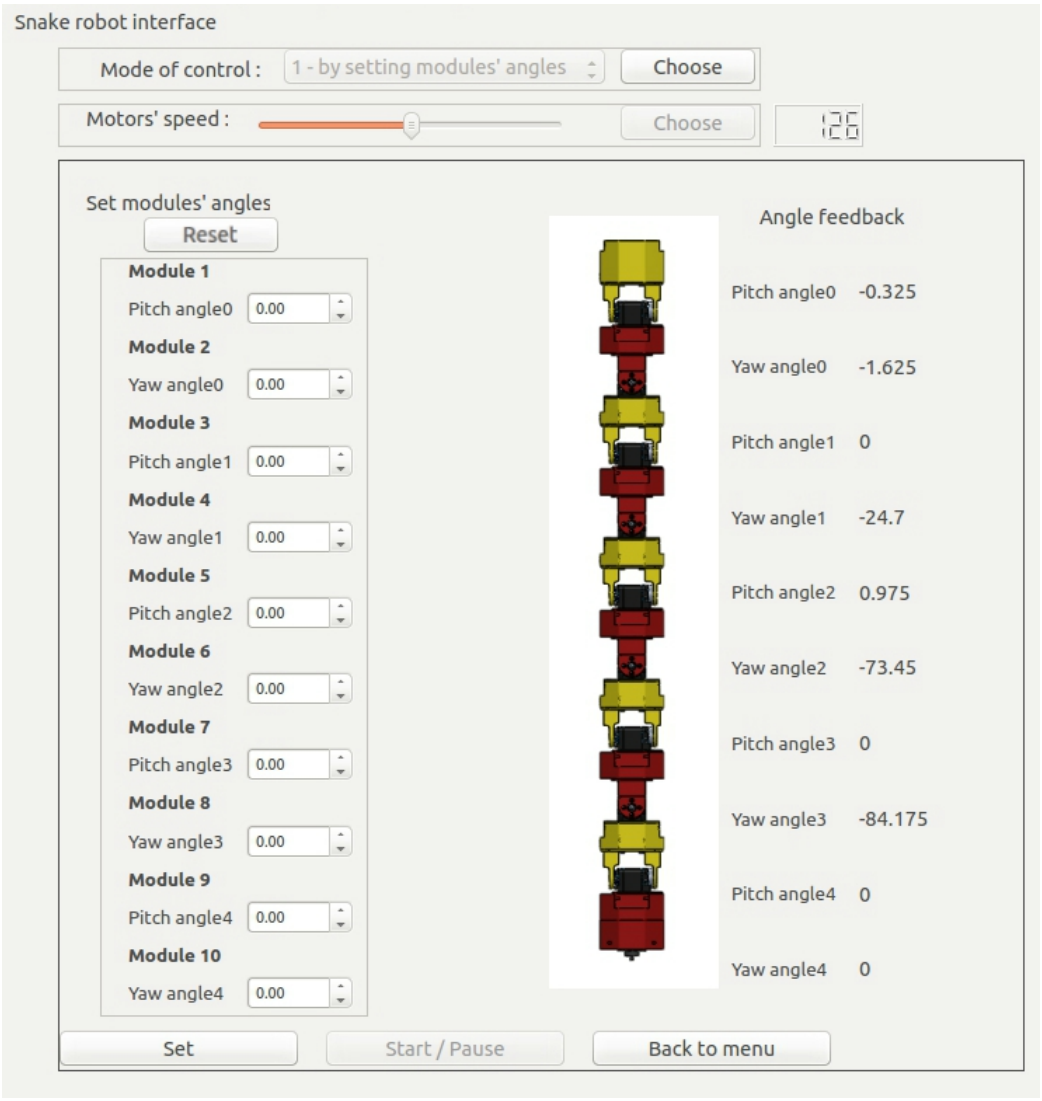
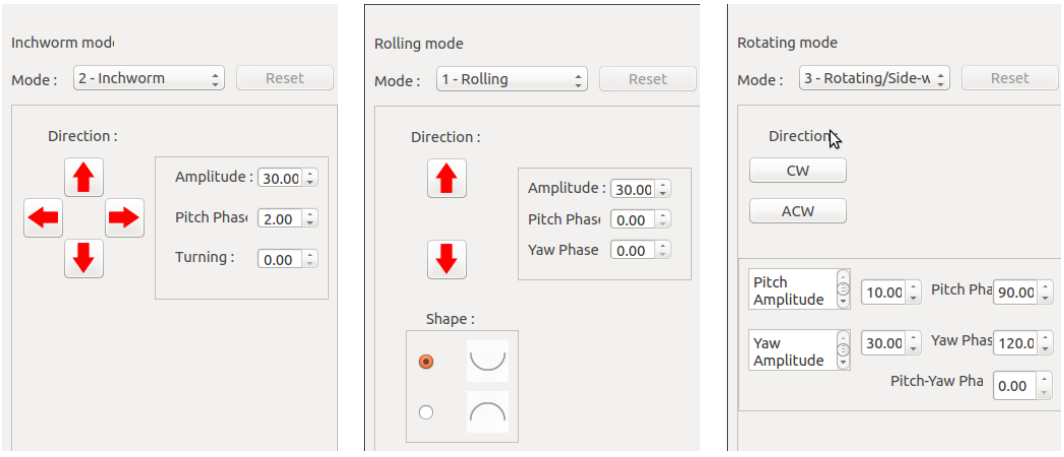


Figure 3.22: Feedback joint angles and buttons in Joint Control mode interface.



(a) **(b)** **(c)**

Figure 3.23: Control interfaces for (a) linear progression gait, (b) rolling gait, (c) rotating gait and sidewinding gait.

3.6 Summary

The design of the modular snake robots can be divided into mechanical, electronic and software aspects. The mechanical aspect of the robot started with the design of the module, and was finalised with four modular prototypes. Out of the four modular prototypes, only the two last modular prototypes were used to build prototype modular snake robots. They were manufactured using a low-cost 3D printer with PLA as the printing material. The third modular prototype had many weaknesses such as joint failures, large bending radius, and difficulty of assembly which the fourth modular prototype managed to resolve.

The electronics design for the first prototype snake robot was very simple with the use of the TTL serial communication method. This method allowed all the smart servo motors, Herkulex DRS-0101, to communicate with the main microcontroller, Atmega 2560. In the second prototype snake robot, a microcontroller, Teensy LC, and an IMU, RS485 board, were added to the design. The Teensy LC reads the outputs from the IMU via I^2C and sends the data to the main microcontroller, Teensy 3.2, via RS485 communication. The TTL serial communication method was still used in the second prototype snake robot. The TTL communication could send and receive data from smart servos in the ten modules up to 10 Hz, and the snake robot could perform locomotion smoothly at this frequency.

The program developed for the modular snake robots, which is based on ROS, controls the robot's locomotion with a GUI. The GUI, which was built using the Qt-based framework, runs in parallel with the motion computation task in the program as two ROS nodes. The GUI was designed in a way that it is intuitive for a user to control the robot locomotion in various circumstances and environments. However, it could not switch gait smoothly; instead, it straightened the robot when the gait mode was changed.

Chapter 4

Optimal Gait Parameter Control

4.1 Introduction

The modular snake robot described in Chapter 3 is capable of performing 2D and 3D locomotion. Nevertheless, it cannot generate movement with lateral undulation gait due to the absence of passive wheels in the design that constrains the movement of the module in the longitudinal axis. Gaits such as linear progression, turning, rolling, rotating and sidewinding can be generated using the sinusoid function generator.

The gait parameters of the sinusoidal function generators influence the external motion generated by the robot. However, depending on the type of gait, the increase in the speed of the motion is limited by the torque of the servos. The increase in the height of the robot may cause the snake robot to lose its balance during locomotion.

Since the modular snake robot used in this thesis consists of four yaw joints and five pitch joints, the robot may not behave the same way as other snake robots from previous studies with the same gait parameter control. The robot may require different conditions of gait parameters to ensure that the external motion generated by the snake robot is efficient while it remains balanced during the locomotion.

This chapter covers the performance of the optimal joint control of the snake robot and presents optimal gait parameters to perform specific gaits for the snake robot in the V-REP simulator and during the physical trials of the robot. The effects of using various gait parameters in gaits are also discussed.

4.2 Gait Pattern Generation

In this chapter, the locomotion of the modular snake robot used as the robotic platform is generated using sinusoid function generators as follows:

$$\varphi(t)_{i,p} = A_p \sin(\omega_p t + \psi_p(i-1)) \quad i \in \{1...M_p\}$$
$$\varphi(t)_{i,y} = A_y \sin(\omega_y t + \psi_y(i-1) + \psi_{py}) + O_y \quad i \in \{1...M_y\}$$

where i is the module number, $\varphi(t)_{i,p}$ and $\varphi(t)_{i,y}$ are the rotation angles of pitch and yaw modules respectively, ω_p and ω_y control the speed of the locomotion speed, and ψ_p and ψ_y control the shape of the snake robot. ψ_{py} sets the phase difference between the yaw and the pitch function generators. O_y is the constant angle added to the yaw function generator. The subscripts p and y indicate that the parameters in the pitch and the yaw function generators are independent. A, ψ and ω without subscript indicate that they are equal in both the pitch and yaw function generators. Setting the parameters for the pitch and the yaw generators differently generates either stable or unstable locomotion. Unstable locomotion is defined when movement generated by the snake robot is not useful.

4.3 Joint Control of the Modular Snake Robot

4.3.1 Peak Angular Velocity of the Joint

In theory, the peak angular velocity of the Herkulex Servo (DRS101) while rotating the joint according to the sinusoid function generators can be obtained by using the time derivatives of the sinusoid function generators. The peak angular velocity, $\dot{\varphi}_{peak}$, can be expressed as shown in Equation 4.1. The relationship between ω and A was computed as presented in Figure 4.1 by using the equation where $\dot{\varphi}_{peak}$ was assumed to be a constant. To ensure that $\dot{\varphi}_{peak}$ does not exceed the maximum rotation speed of the motor, $\dot{\varphi}_{peak}$ was set to 6.28 rad/s.

$$\dot{\varphi}_{peak} = A \cdot \omega \tag{4.1}$$

Based on Figure 4.1, as A increases, ω that can be set in function generators decreases. Although $\dot{\varphi}_{peak}$ would still be below 6.28 rad/s when $A < 30°$, using ω that exceeds 6.28 rad/s would cause the snake robot to vibrate instead of generating a smooth motion. This vibration was caused by aliasing in the sinusoid waveform as fewer angles could be interpolated with the interpolation rate of 10 Hz, which was limited by the TTL communication of the robot.

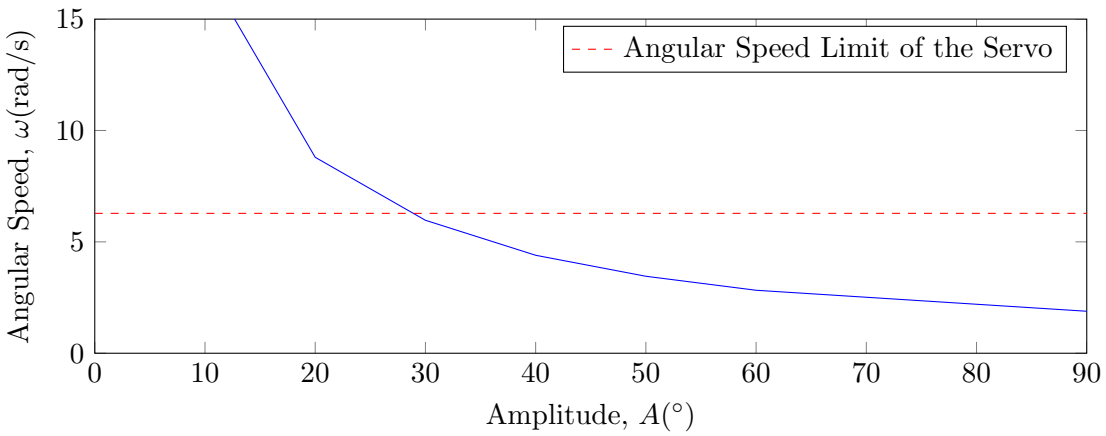


Figure 4.1: The maximum ω at different level of A without load.

4.3.2 Optimisation of the Joint Control

The optimal P, I and D gains for a single servo was tuned by trial and error. The performance of the gains was tested with unit step responses of 0° and ±9.425° based on the requirements of the step-response characteristics as presented in Table 4.1. The servo was first tested by rotating both the first pitch and the head module from side to side. To test the performance of the PID gains under a load, the robot was then rotated as shown in Figure 4.2 so that the first yaw module would lift and drop both the first pitch and the head modules.

Table 4.1: The requirements of the step-response characteristics for the PID control of the servo.

Characteristic	Requirement
Min. rise-time (s)	< 0.1
Min. settling time (s)	< 0.1
Max. Overshoot error (%)	< 15
Max. Steady-state error (°)	< 1

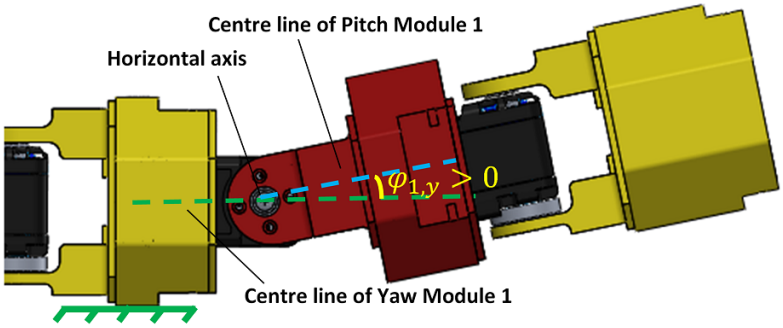


Figure 4.2: The setup to test the performance of the PID controller under a load.

While tuning the gains, the time rise decreased as P gain was increased. However, the overshoot error increased and the system started to be underdamped and settled slowly with the increase of P gain. To reduce the overshoot error and the settling time, D gain was increased. Increasing I gain did not improve the performance of the controller, instead it caused the system to be more unstable. It was found that the optimal P, I, and D gains were 480, 0 and 8191 respectively. These gains are only applicable for the in-built PID control of the Herkulex DRS-0101 servo and may not work with other types of servos. The characteristics of the unit step response of the motor under no load and under the load are as presented in Figure 4.3 and Figure 4.4 respectively.

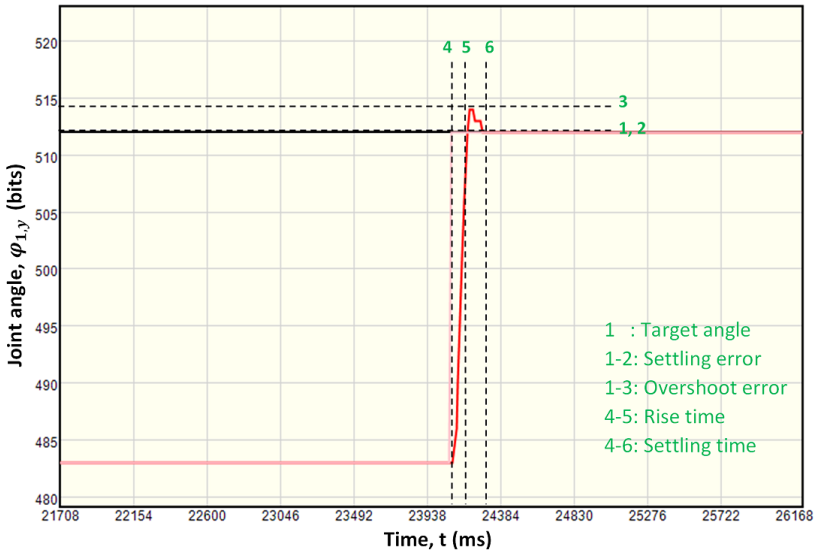


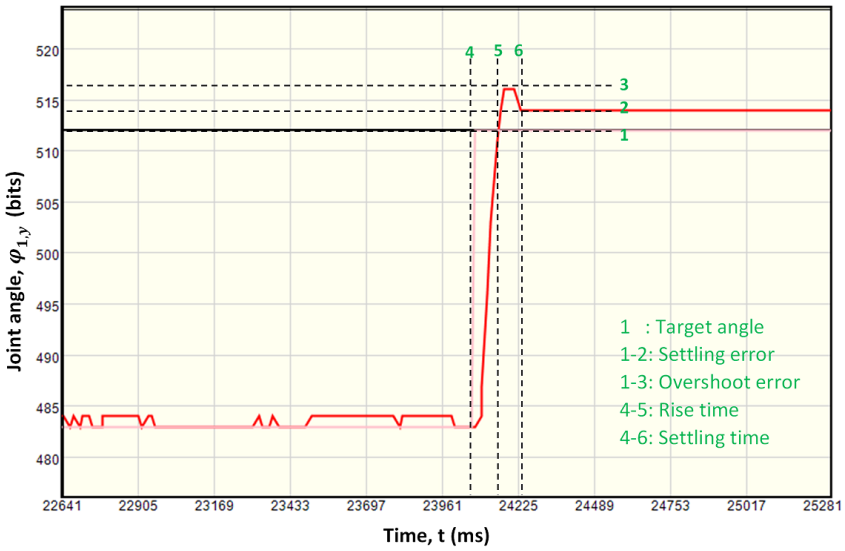
Figure 4.3: The characteristics of the optimal PD control output under no load.

Based on Table 4.2, the optimal PD control met most of the requirements, except for the minimum settling time. Although the minimum settling time did not meet the requirement, it would not be an issue to control the joint as the rotation angle would have reached approximately to the target angle, and the next target angle would be instructed after 0.1 seconds.

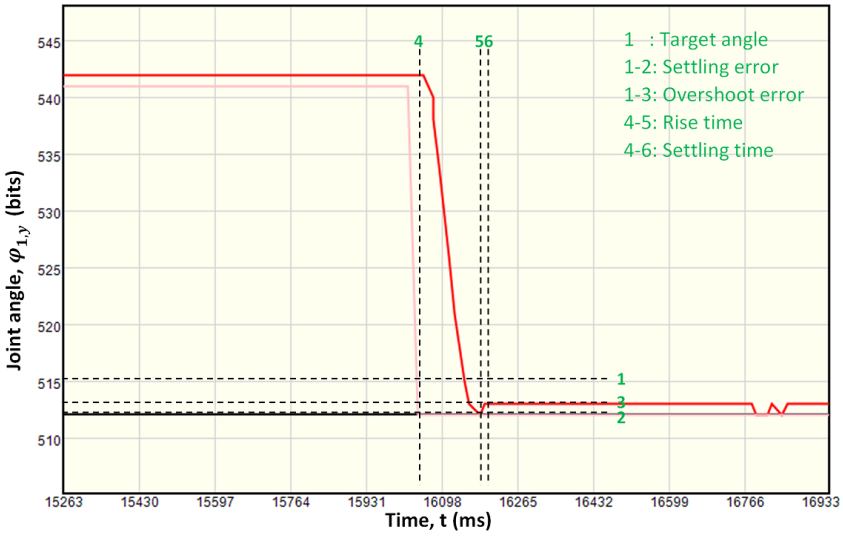
Table 4.2: The performance of the optimal PD control of the Herkulex servo.

Characteristic	No load	Lowering load	Lifting load
Min. rise-time (s)	0.085	0.098	0.126
Min. settling time (s)	0.169	0.151	0.145
Max. Overshoot error (%)	6.7	13.8	0.0
Max. Steady-state error (°)	0.0	0.325	0.325

The optimal PD control appeared to have a larger maximum steady-state error under the load. This error might be due to the inadequacy of the servo torque. This effect can be seen in Figure 4.4a and Figure 4.4b. When the load was lifted from the bottom ($\varphi_{1,y} = -9.425°$), the joint reached to the target angle ($\varphi_{1,y} = 0°$) but then dropped back to below the target angle ($\varphi_{1,y} < 0°$). On the other hand, the joint angle further travelled below the target angle ($\varphi_{1,y} = 0°$) when the load was lowered from the top ($\varphi_{1,y} = +9.425°$), resulting in a large overshoot error. The inadequacy of the servo torque would cause a similar behaviour for a larger A but with a maximum steady-state error.



(a) Lowering the load



(b) Rising the load

Figure 4.4: The characteristics of the optimal PD control output under a load (note that $\varphi_{1,y}(°) = -[\varphi_{1,y}(\text{bits}) - 512] \cdot 0.325$, meaning that $\varphi_{1,y}(°) > 0°$ when $\varphi_{1,y}(\text{bits}) < 512$).

4.3.3 Performance of the Optimal PD Control in Gaits

The performance of the optimal PD Control was further studied by performing various gaits such as lateral undulation, linear progression, sidewinding, rotating and rolling. The gait parameters used to generate the gaits are presented in Table 4.3. The feedback joint angles of the modular snake robot were recorded for 12 seconds from the ROS program.

Table 4.3: The gait parameters used for trials to test the performance of the PD control.

Gait	Trial	$A_p(^\circ)$	$A_y(^\circ)$	ω_p(rad/s)	ω_y(rad/s)	$\psi_p(^\circ)$	$\psi_y(^\circ)$	$\psi_{py}(^\circ)$
Lateral	1	0	30	0	0.5π	0	144	0
Undulation	2	0	30	0	2π	0	144	0
Linear	3	30	0	π	0	144	0	0
Progression	4	30	0	2π	0	144	0	0
Sidewinding	5	10	30	0.5π	0.5π	144	144	0
	6	10	30	π	π	144	144	0
Rotating	7	10	30	0.5π	0.5π	90	120	0
	8	10	30	π	π	90	120	0
Rolling	9	30	30	π	π	0	0	90
	10	30	30	2π	2π	0	0	90

The performance of the control was analysed by determining the mean of the errors in the feedback joint angle, $\bar{e}_\varphi$, and the difference between the upper and the lower peak errors, e_{pp}. $\bar{e}_\varphi$ and e_{pp} were calculated using Equation 4.2 and Equation 4.3 respectively.

$$\bar{e}_\varphi = \sum_{k=1}^{n} |e_\varphi(t_{top})| + \sum_{k=1}^{n} |e_\varphi(t_{bottom})| \tag{4.2}$$

$$\bar{e}_\varphi = \sum_{k=1}^{n} |e_\varphi(t_{top})| - \sum_{k=1}^{n} |e_\varphi(t_{bottom})| \tag{4.3}$$

As shown in Figure 4.5, $e_\varphi(t_{top})$ and $e_\varphi(t_{bottom})$ are the errors at the top upper and the lower peak of the feedback joint angle. t_{upper} and t_{lower} is the time when the φ reaches the upper and the lower peaks respectively. n is the number of peaks (samples) used for averaging. The feedback joint angles of most joints have similar outcomes. For most trials, the feedback joint angle of the third pitch joint of the robot was examined, but the angle of the third yaw joint was examined instead for Trial 1 and 2 (pitch joints are not used in lateral undulation gait). A higher n would give a more accurate outcome. Since the feedback joint angle does not change drastically, n was set to 2 in this study.

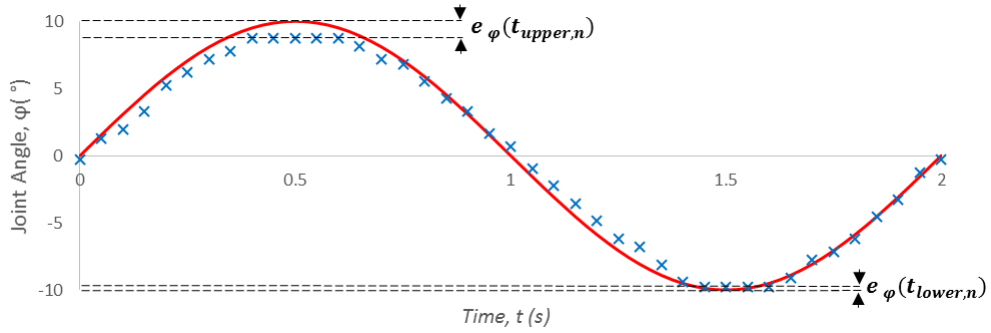


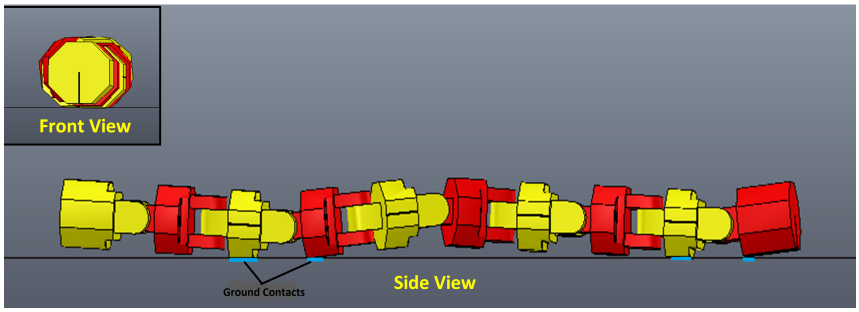
Figure 4.5: The error at the upper and the lower peak of the feedback joint angle.

Table 4.4: Results of the mean of the feedback joint errors obtained from trials.

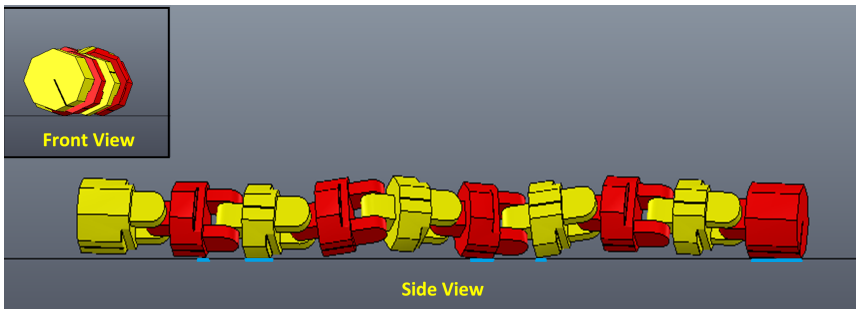
Trial	1	2	3	4	5	6	7	8	9	10
$\bar{e}_{\varphi}$ (%)	0.9	4.7	3.1	6.3	9.1	9.1	1.6	1.6	0.9	7.9
e_{pp} (%)	1.1	0.0	3.2	7.6	6.5	6.5	1.8	1.8	1.1	0.0

Based on Table 4.4, $\bar{e}_{\varphi}$ of Trial 2, 4 and 10 are relatively larger than those of other trials because $\dot{\varphi}_{peak}$ is approximately equal to the maximum rotation speed of the servo when $A = 30°$ and $\omega = 2\pi$ rad/s as shown in Figure 4.1. Trial 2 and Trial 4 appear to have significantly small e_{pp} because there was minimal load exerted during the lateral undulation and the rolling gaits.

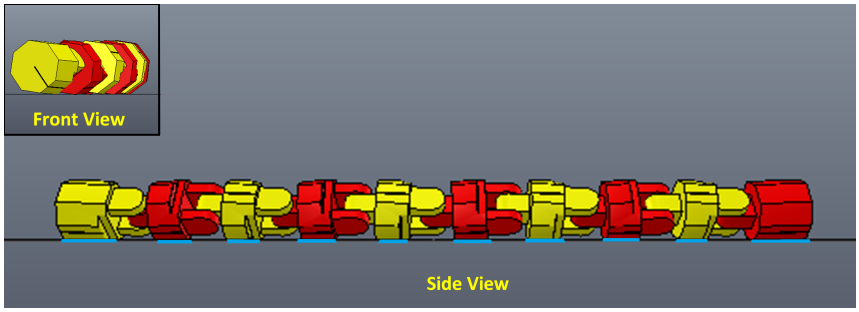
Although Trial 5 and 6 were set with $\omega < 2\pi$ rad/s, both $\bar{e}_{\varphi}$ and e_{pp} appear to be significantly large. These large errors might be due to the large amount of torque exerted to the joints of the robot during the locomotion. The amount of torque exerted to the joints depends on the number of ground contacts and the orientation of the snake robot. Based on the 4.6, the sidewinding gait has the smallest surface area of ground contacts and the robot's orientation is upright, resulting in the large amount of torque exerted to the joints during the locomotion. The rotating and the rolling gaits have large surface of ground contacts that distribute the torque exerted to the joints more evenly. The tilted robot during these gaits also allow the torque to be distributed evenly among the pitch and the yaw modules. The load from the robot's body does not exert torque to the yaw joints, and this can be seen from the errors in the joint angles of the lateral undulation gait in Trial 1 and 2, as presented in Table 4.4. This explains why the errors obtained for the rotating gait and the rolling gait are smaller than that for the sidewinding gait. Figure 4.7 shows the effect of the torque on the motion of the third pitch joint while the robot performed the sidewinding gait (Trial 5) and the rotating gait (Trial 7).



(a) Sidewinding



(b) Rotating



(c) Rolling

Figure 4.6: The surface area of ground contacts and the orientation of the snake robot during the sidewinding, rotation and rolling gaits.

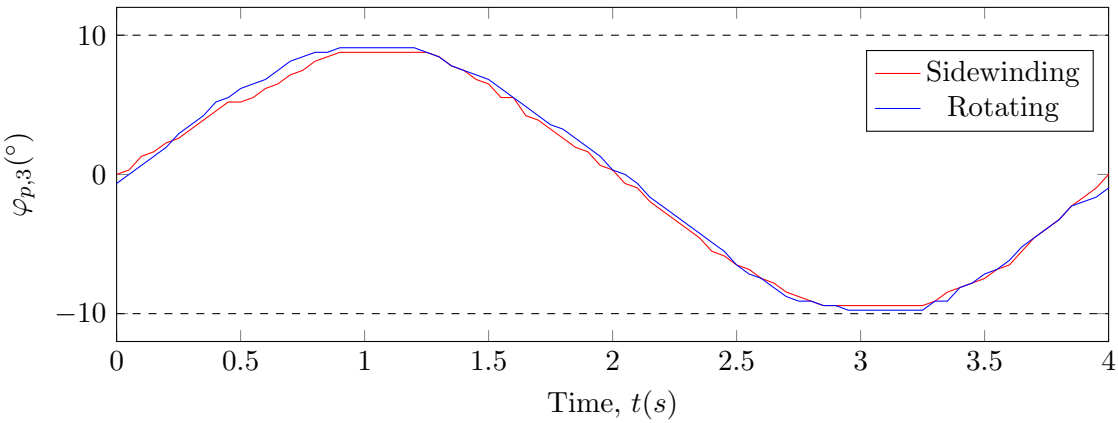


Figure 4.7: The joint feedback angles of the third pitch module with flat top peaks from Trial 5 and 7.

4.4 Simulation Setup

The V-REP simulator (Version 3.2.2, Coppelia Robotics) [90] was used to simulate the locomotion of the snake robot with ten modules. The purpose of using the simulator was to prevent damaging the actual robot if the locomotion became unstable. Furthermore, it was quicker and easier to set up the environment to test and validate the locomotion of the snake robot.

To ensure that the simulator ran smoothly and efficiently, a simplified version of the modular snake robot discussed in Chapter 3 was modelled and exported to the simulator as shown in Figure 4.8. Before importing the 3D model of the robot to the simulator, the model was converted into URDF format. The imported robot model was then set to have the same mass properties as the actual robot to ensure that the locomotion of the robot in the simulation would behave in the same manner as in the real world with minimum errors. The same time step used in the actual robot was set to control the robot in the simulator.

A GUI was designed to control and record the locomotion of the snake robot. The ODE physics engine was used to run the simulation as the license was free and the engine simulated both rigid body dynamics and collision detection. The details of the simulator settings are shown in Figures A.1-A.5 (see Appendix A). Since the V-REP simulator uses both dynamics and kinematics, it is important to be aware that the dynamics of the snake robot's locomotion may not behave the same as in the real world.

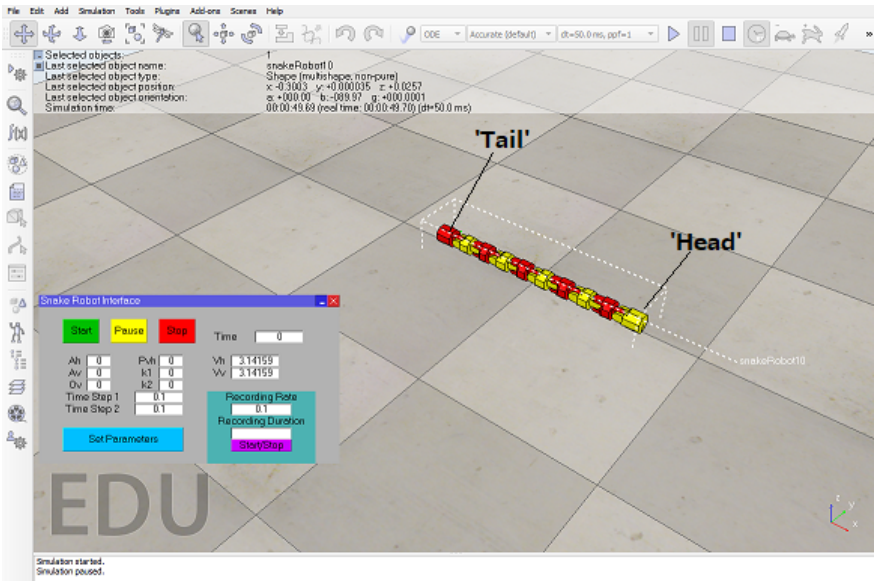


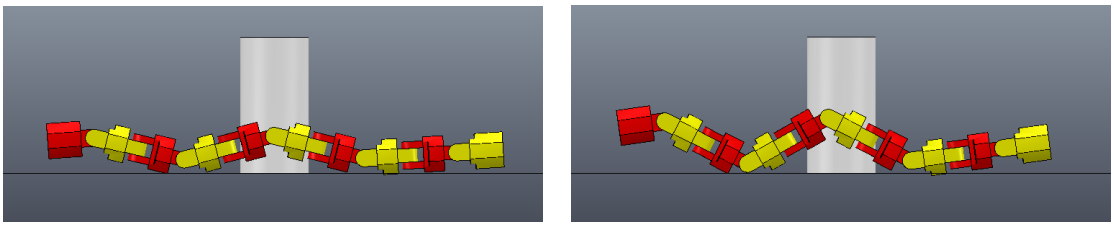
Figure 4.8: The simplified snake robot consisting of ten modules in the V-REP simulator.

4.5 Optimal Gait Parameters

4.5.1 Linear Progression Gait

Linear progression gait is considered as 2D locomotion as the robot travels in a single axis only. This gait can be generated by using only the pitch function generator, setting the gait parameters in the pitch function generator to non-zeros, similarly to the lateral undulation gait which instead uses the yaw function generator. The linear progression gait can be modified to become the turning gait by setting O_y to a non-zero. Unlike the lateral undulation gait, the linear progression gait does not require special frictional properties to generate movement as long as the modules can grip a surface with minimal slip. There are three main aspects which can be controlled in the linear progression gait: the height, direction, and speed of locomotion.

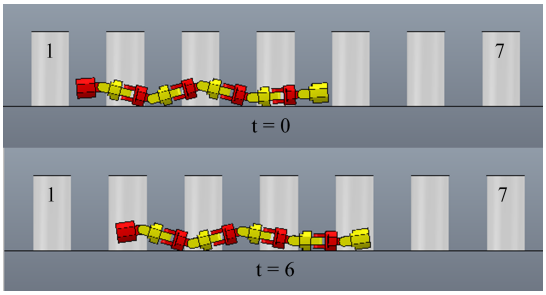
Controlling the height of a snake robot is important to allow the robot to crawl or climb in-between gaps or under obstacles, and to control its balance. A pillar was used as the reference for the height of the robot in the simulation. Increasing A_p, increases the height of the robot as shown in Figure 4.9. Increasing A_p also increases the speed of the locomotion of the robot. To validate this effect, the robot was tested in the simulation to travel from one pillar to another within 6 seconds. The robot with $A_p = 30°$ travelled until the fifth pillar, whereas with $A_p = 50°$, it travelled closer to the sixth pillar as shown in Figure 4.10.



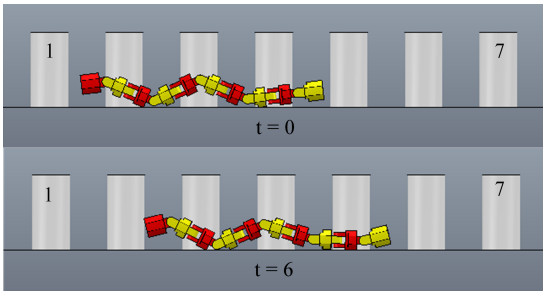
(a) $A_p = 30°$ **(b)** $A_p = 50°$

Figure 4.9: The height of the snake robot while performing linear progression gait with $\psi_p = 144$ but with two different A_p.

Another gait parameter that increases the height of the snake robot is ψ_p. Unlike A_p, ψ_p needs to be decreased to increase the height of the robot as shown in Figure 4.11. The smaller the ψ_p, the higher the height of the robot and the faster the speed of its locomotion. The effect ψ_p on the speed was validated with the same test set up as conducted previously in the simulation.

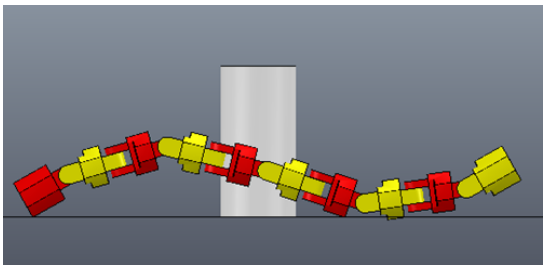


(a) $A_p = 30°$ and $\psi_p = 144°$

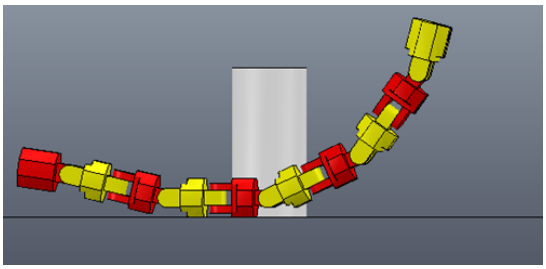


(b) $A_p = 50°$ and $\psi_p = 144°$

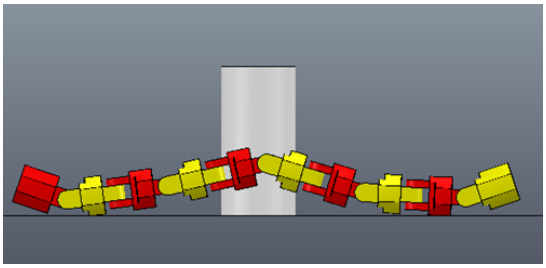
Figure 4.10: The distance travelled by the robot while performing linear progression gait with two different magnitudes of A_p in 6s.



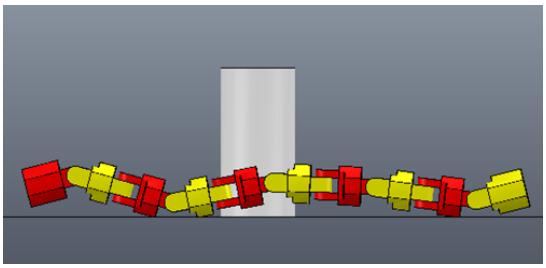
(a) $A_h = 30°$ and $\psi_p = 36°$



(b) $A_h = 30°$ and $\psi_p = 72°$



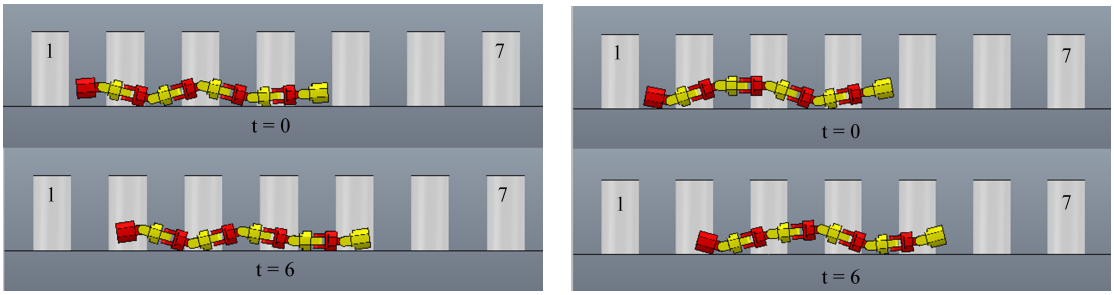
(c) $A_h = 30°$ and $\psi_p = 108°$



(d) $A_h = 30°$ and $\psi_p = 144°$

Figure 4.11: The height of the snake robot while performing linear progression gait with different magnitudes of ψ_p at $t = 0$.

It was found that the robot with smaller ψ_p travelled further as shown in Figure 4.12. As ψ_p approaches 180°, the motion would generate less displacement and start to behave similarly to an underdamped spring system undergoing simple harmonic motion when $\psi_p = 180$. The amplitudes generated by the function generator for consecutive modules would have the smaller difference in the magnitudes but with the opposite sign as ψ_p approaches 180°. Unless the snake robot is crawling or climbing between gaps, the increase in the height of the snake may cause it to lose its balance during locomotion. Based on the test conducted with the actual snake robot, it was found that setting $A_p > 60°$ and $\psi_p < 120°$ was impractical as the robot became unbalanced.



(a) $A_p = 30°$ and $\psi_p = 144°$ **(b)** $A_p = 30°$ and $\psi_p = 108°$

Figure 4.12: The distance travelled by the robot while performing linear progression gait with two different magnitudes of ψ_p in 6s.

The direction of the locomotion of the snake robot can be changed in many ways such as setting ψ_p beyond 180° and modifying the sign of ω_p. Changing the sign of ω_p as in Figure 4.13 is recommended because the shape of the robot will be inverted when $\psi_p > 180°$.

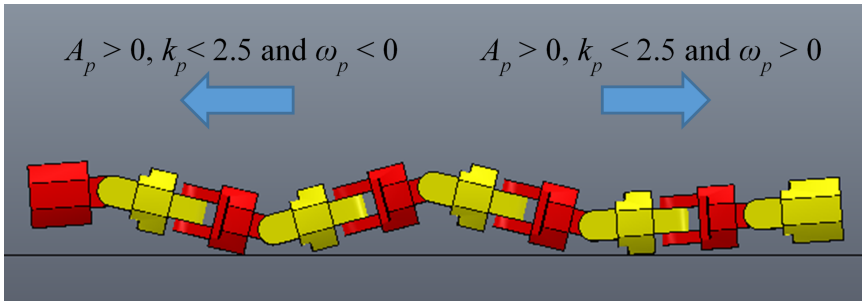


Figure 4.13: The direction control of linear progression gait.

The increase in ω_p would also increase the speed of its locomotion. Nevertheless, it is important to be aware of maximum torque of the servos when increasing it. As a rule of thumb, it is safe to assume that the maximum driving torque should not exceed 0.6 Nm as discussed in Chapter 3.

4.5.2 Turning Gait

The turning gait in this study was generated by forming the body of the snake robot into an arc while performing a linear progression gait by setting O_y to a non-zero value. The height, the speed, and the direction of turning gait behave in similar ways to linear progression gait. The main differences of turning gait as compared to linear progression gait are the trajectory path of the locomotion and the effect of the height on the stability of the snake.

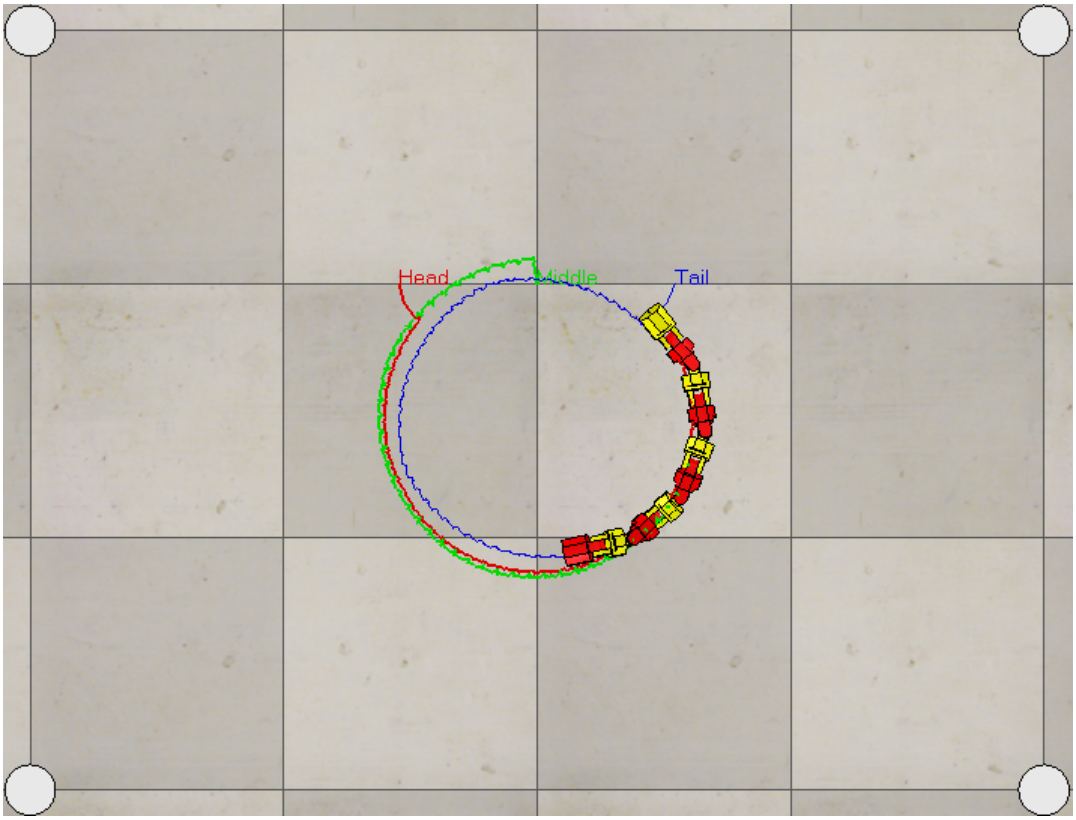
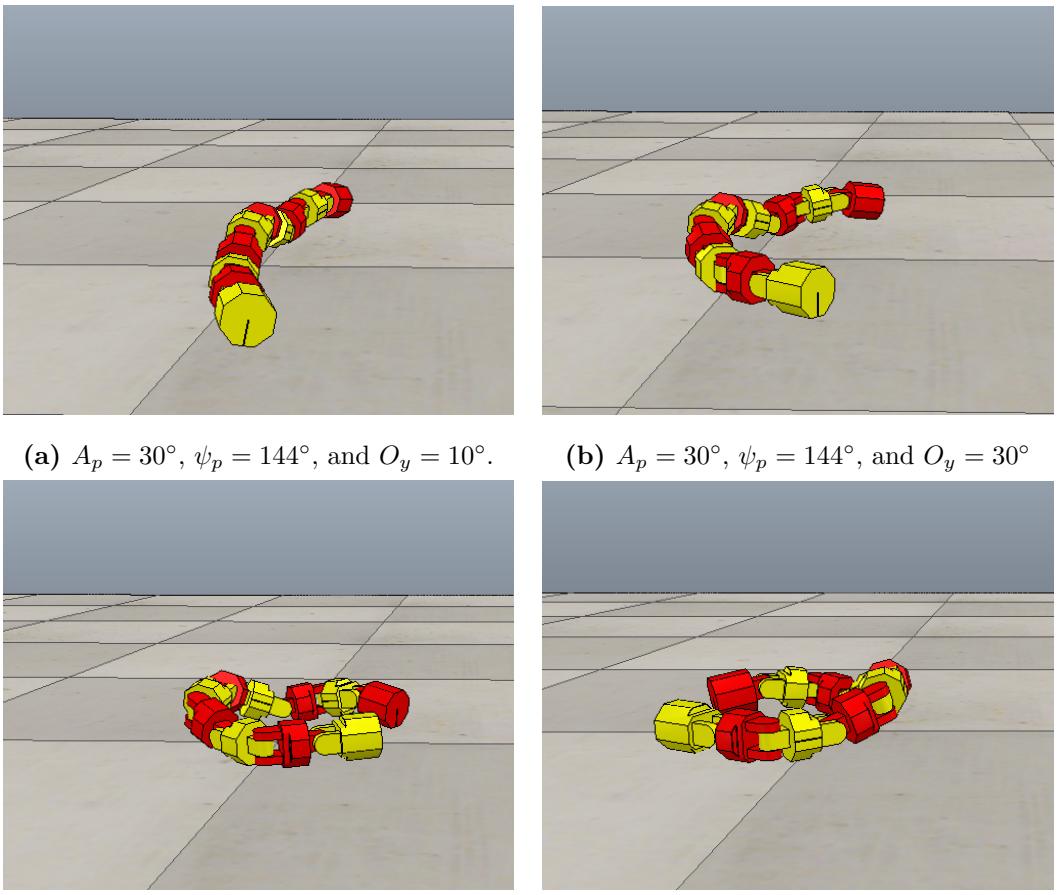


Figure 4.14: The motion trajectory of turning gait.

Instead of moving in a straight trajectory path, the turning gait moves according to the path of the body arc of the snake robot which is formed when $O_y \neq 0$ as shown in Figure 4.14. Increasing O_y decreases the radius of the arc; due to fixed total length of the snake, the snake's body shape will become more tightly curved as shown in Figure 4.15. Based on Figure 4.15, the body curve is formed in the opposite direction when the sign of O_y changes. The smaller the radius of the robot's body, the more stable it becomes; thus, its height can be increased slightly. Based on the test conducted on the actual snake robot, it was found that the maximum O_y for the robot was $\pm 68°$. Setting O_y greater than this angle caused the head and the tail modules to collide with each other.



(a) $A_p = 30°$, $\psi_p = 144°$, and $O_y = 10°$.

(b) $A_p = 30°$, $\psi_p = 144°$, and $O_y = 30°$

(c) $A_p = 30°$, $\psi_p = 144°$, and $O_y = 60°$

(d) $A_p = 30°$, $\psi_p = 144°$, and $O_y = -60°$

Figure 4.15: The radius and direction of the curvature of the robot while performing turning gait with different magnitudes and signs of O_y.

4.5.3 Sidewinding Gait and Rotating Gait

Sidewinding and rotation gaits are generated by the superposition of pitch and yaw waveform, and they mainly depend on ψ_p and ψ_y. In both gaits, it is important to ensure that $A_y >= A_p$ to maintain the snake robot's balance. The increase in A_y would increase motion speed of the snake robot. Increasing A_y may help the snake robot to climb over small obstacles but with the risk of the robot losing its balance.

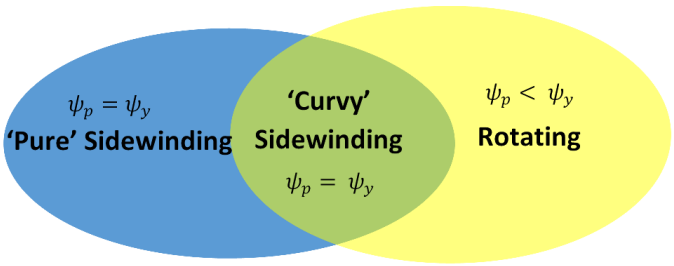


Figure 4.16: The intercept between the sidewinding and rotation gait in the locomotion of the modular snake robot.

While performing the sidewinding gait with the modular snake robot, the external motions generated by the sidewinding gait with certain gait parameters were a mixture of that of rotating gait. The 'curvy' sidewinding was generated with the same conditions of ψ_p and ψ_y as a 'pure' sidewinding as shown in Figure 4.16. The rotational motions of this gait might have been generated when ground contact tracks (GCTs) are not parallel as shown in Figure 4.17 [91]. GCTs might not be parallel due to the lack of the number of module of the robot. A 'pure' sidewinding gait is defined when the robot is moving straight with a minimal change to the orientation of the robot. On the other hand, a 'pure' rotation gait is defined when the snake robot rotates about its centre of mass with a relatively small radius of circular motion of the middle joint of the snake robot. The motions of these gaits are illustrated in Figure 4.18.

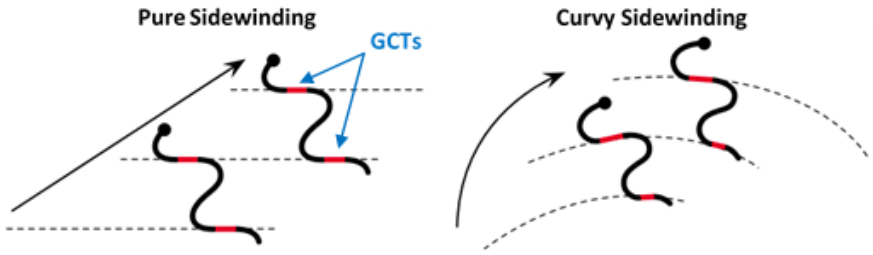
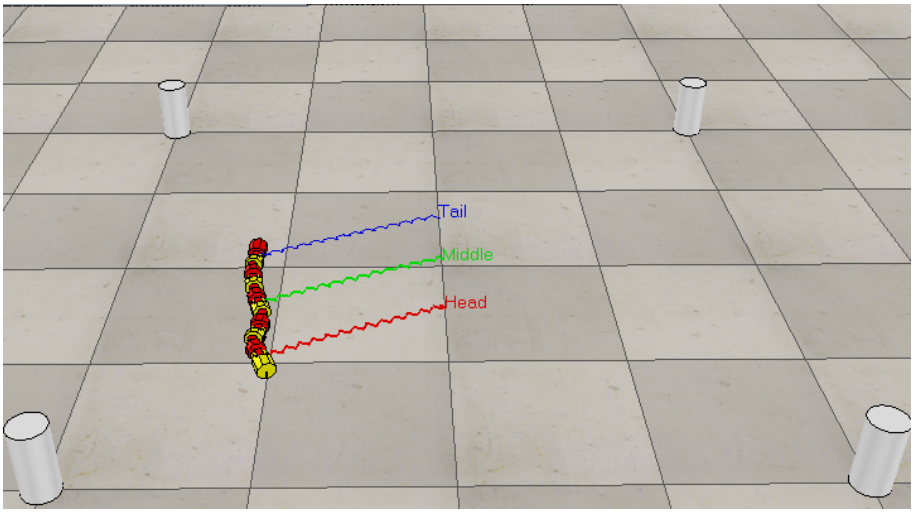
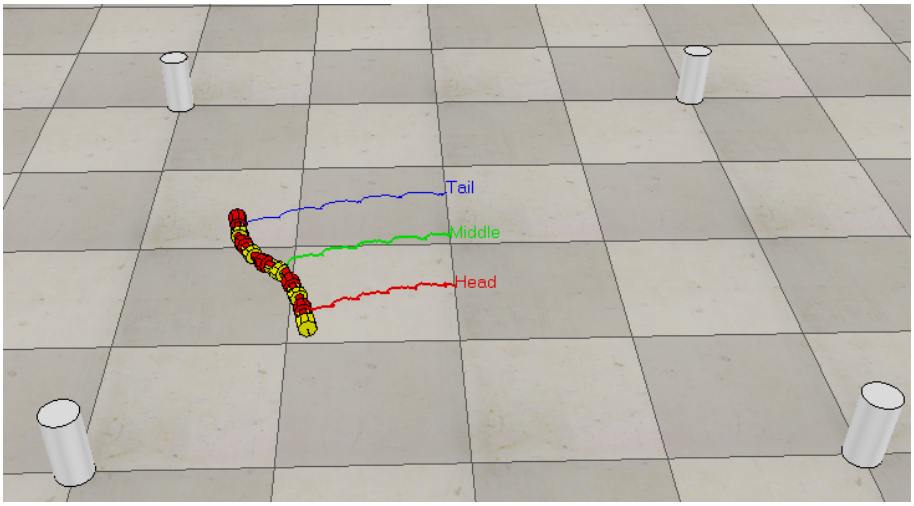


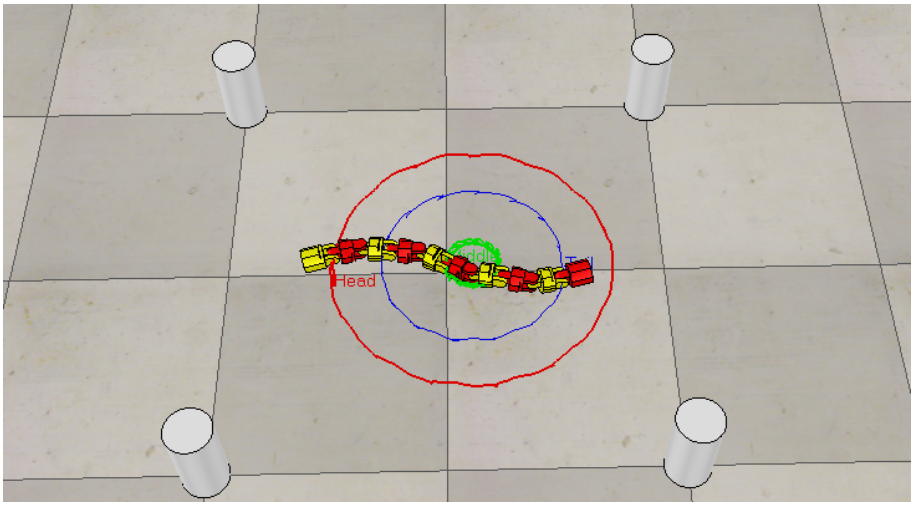
Figure 4.17: The effect of the GCTs on the motion trajectory of the sidewinding gait (adapted from [18]).



(a) $A_p = 10°$, $A_y = 30°$, $\psi_p = \psi_y = 144°$, $\psi_{yp} = 0°$, and $\Delta t = 10s$



(b) $A = 10°$, $A_y = 30°$, $\psi_p = \psi_y = 100°$, $\psi_{yp} = 0°$, and $\Delta t = 10s$



(c) $A = 10°$, $A_y = 30°$, $\psi_p = 90°$, $\psi_y = 120°$, $\psi_{yp} = 0°$, and $\Delta t : 20s$

Figure 4.18: Motion of the snake robot while performing three different types of sidewinding gaits.

To generate sidewinding gait, the function generators are set with $\psi_p = \psi_y$ [21]. In this study, the shape of the motion trajectory for the gait could not be controlled because the robot's shape was not set to fit according to the backbone curve model as proposed by Burdick et al. [91]. Using the modular snake robot, a 'pure' sidewinding gait could easily be generated when $\psi_p = \psi_y = 144°$. Otherwise it would be a 'curvy' sidewinding instead as the GCTs become less parallel with each other. Figure 4.19 shows the four conditions of gait parameters which are required to change the direction of the motion of sidewinding gait. The direction of the motion of sidewinding gait in the y-axis can be controlled by setting ψ_{py} in the range of $[0, \psi_p]$. When $\psi_{py} = \frac{\psi_p}{2}$, there will be no motion generated by the snake robot. Motion in the positive and the negative directions of the y-axis are generated when $\psi_{py} > \frac{\psi_p}{2}$ and $\psi_{py} < \frac{\psi_p}{2}$ respectively, but the motion in the x-axis remains the same. To change the motion in the x-axis, the sign of angular speed, ω, needs to be changed.

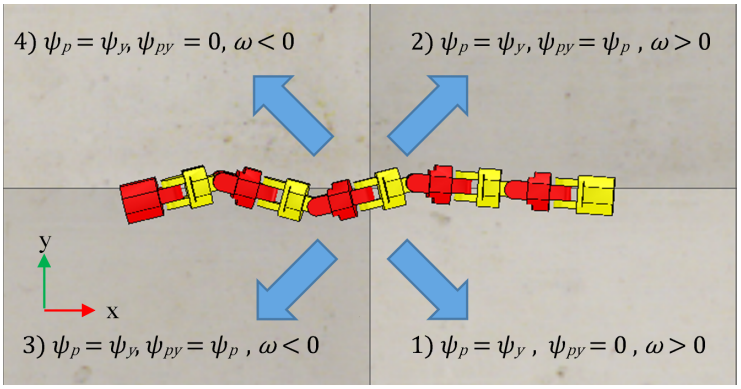


Figure 4.19: The direction of the motion of sidewinding gait under four conditions of gait parameters.

To generate rotating gait, $\frac{\psi_p}{\psi_y}$, needs to be 0.75 and ψ_p needs to be in the range of $[60, 120]$. This ratio is different from the ratio used in the studies [22, 17] which was between 0.5-0.6. The difference in the ratio might be due to the different number of pitch and yaw modules of the robot. Figure 4.18 shows that the middle module of the robot moves in a circular motion. Its radius depends on the ratio of the amplitudes, $\frac{A_y}{A_p}$, and the magnitude of ψ_p. The motion of the 'middle' can be unpredictable when $A_p < 5°$ because larger surfaces of the modules would be touching the ground. In this condition, the robot would rotate less effectively due to friction. As long as $\psi_p > 90$ and $\frac{\psi_p}{\psi_y} = 0.75$ and $\frac{A_y}{A_p} > 5$, the change in the position of the 'middle' would remain small. To change the direction of the rotation, the sign of the angular speed, ω, needs to be changed as shown in Figure 4.20.

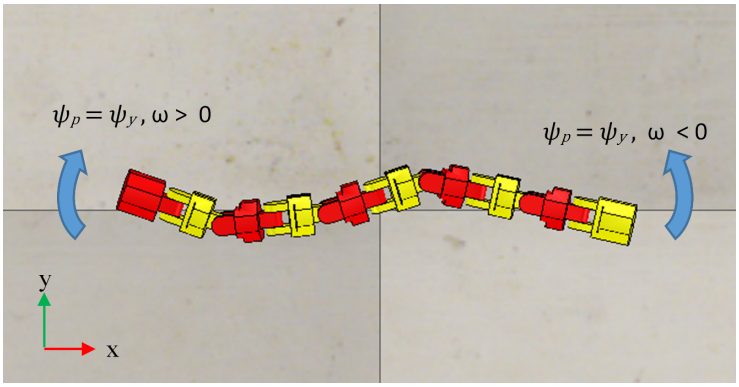
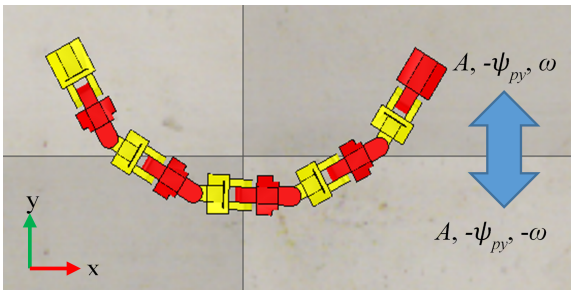


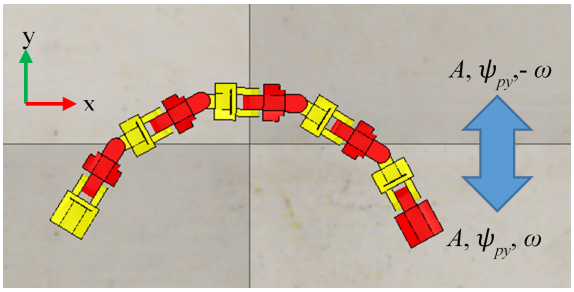
Figure 4.20: The direction of the rotation of rotating gait with different signs of ω.

4.5.4 Rolling

There are two main types of rolling gaits: lateral rolling and pole climbing. The simplest rolling gait in which all the modules of the modular snake robot touch the flat ground is lateral rolling. To generate the gait, the gait parameters need to be set with $\psi_{py} = 90°$, $\psi_p = \psi_p = 0°$and $A_p = A_y$. The signs of the amplitudes, A (for both A_p and A_y), and ψ_{py} affect the direction of the motion and the curvature of the rolling as shown in Figure 4.21.



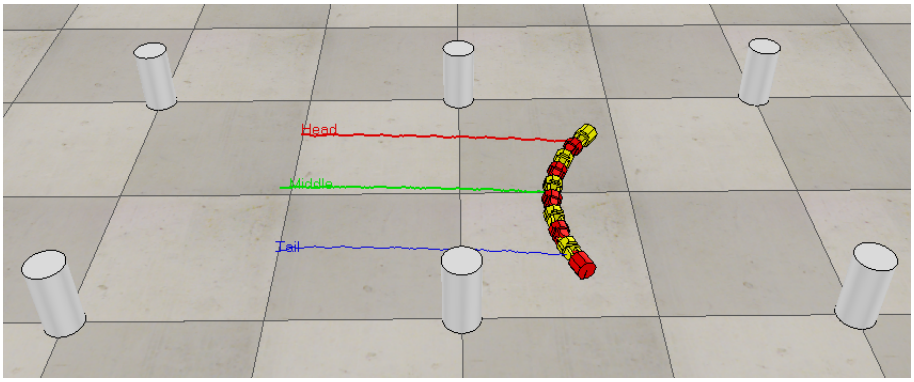
(a) The shape of a snake robot is curving downwards along the y-axis



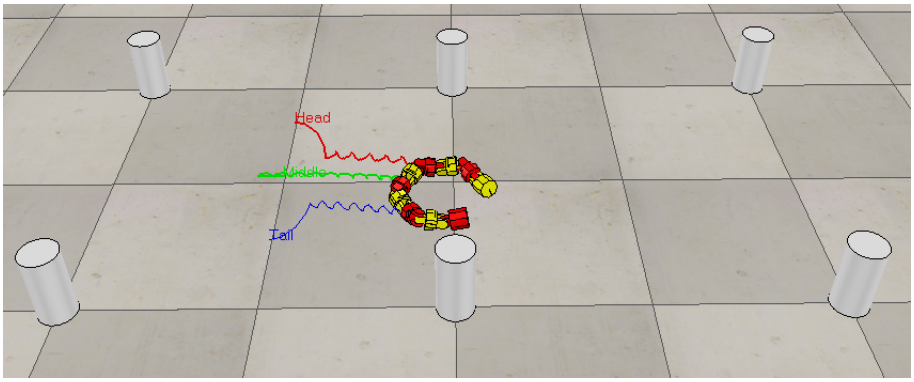
(b) The shape of a snake robot is curving upwards along the y-axis

Figure 4.21: The direction of the curvature of rolling gait with different signs of A and ψ_{vh}.

Figure 4.22 shows that increasing A decreases the radius of the curvature of the snake's body and the internal motion displacement of the robot. The decrease in the internal motion displacement can be explained by examining the vector direction of the joint as presented in Figure 4.23. For a smaller A, most of the motion vectors of the joints are pointing more parallel with the motion vector of the middle joint of the robot. On the other hand, some of the motion vectors are pointing toward each other which cancel out the generated motion displacement as A increases.



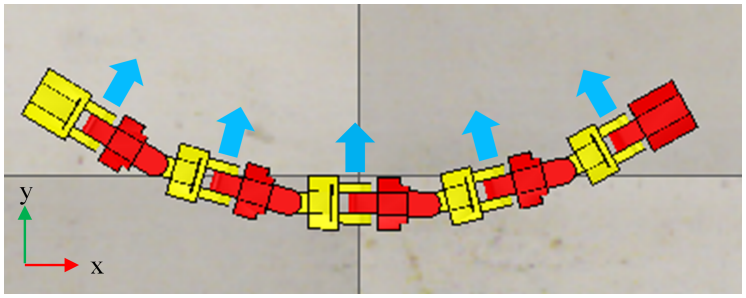
(a) $A = 15°$ and $\psi_{vp} = 90°$



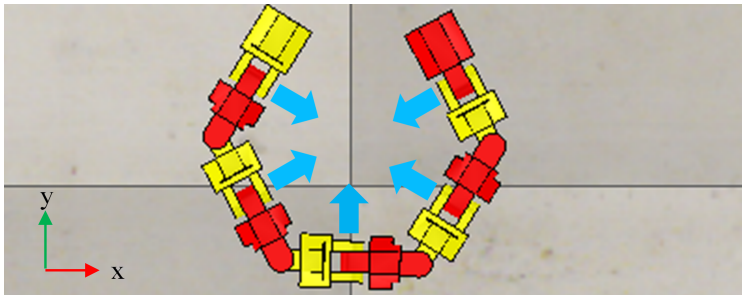
(b) $A = 60°$ and $\psi_{vp} = 90°$

Figure 4.22: The differences in the radius of the curvature of the robot and the distance it travelled for rolling gait with two different amplitudes, A, in 10 seconds.

Although increasing A might not be beneficial for the snake robot to travel on a flat surface, the decrease in the radius of the curvature of the snake would be useful for the snake to clamp. It was found that maximum A for the modular snake robot to perform lateral rolling gait was 68°.



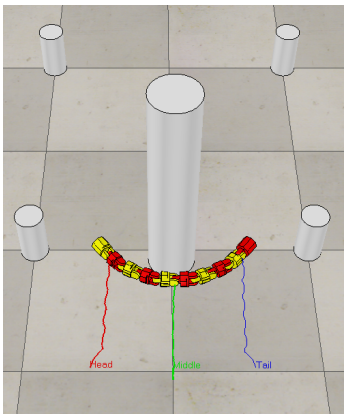
(a) $A = 15°$ and $\psi_{vp} = 90°$



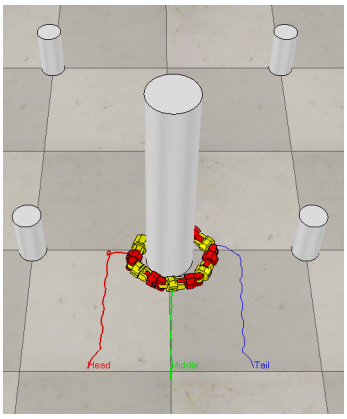
(b) $A = 60°$ and $\psi_{yp} = 90°$

Figure 4.23: The internal motion vectors of the joints with two different amplitudes of A.

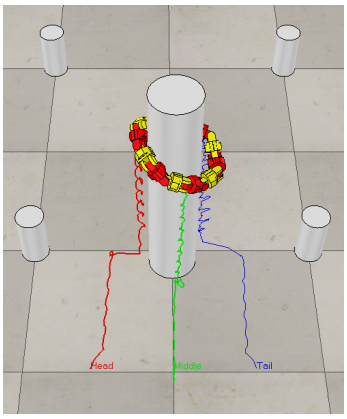
Pole climbing gait is performed when the snake robot clamps itself around the pole and rolls on a pole without slipping off of it. Before the robot could clamp around the pole, it should approach towards the pole with lateral rolling gait. The curvy shape of the robot formed while performing the lateral rolling prevents the robot from deviating away from the pole as the robot hits the pole. A is increased gradually as soon as the robot hits the pole until the robot manages to adequately grip and roll on the pole. These steps are summarised and presented in Figure 4.24.



(a) Lateral rolling



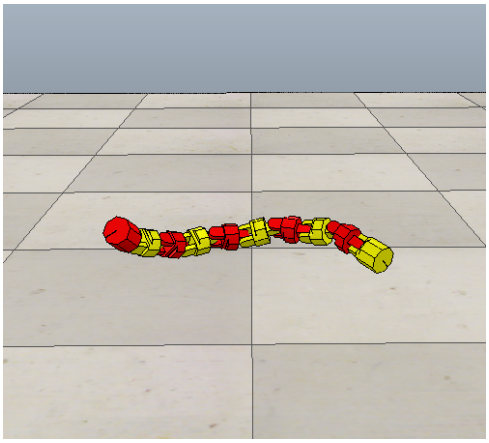
(b) Pole clamping



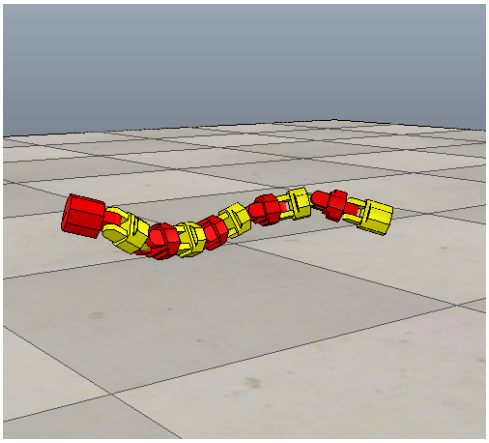
(c) Pole climbing

Figure 4.24: The steps to perform pole climbing.

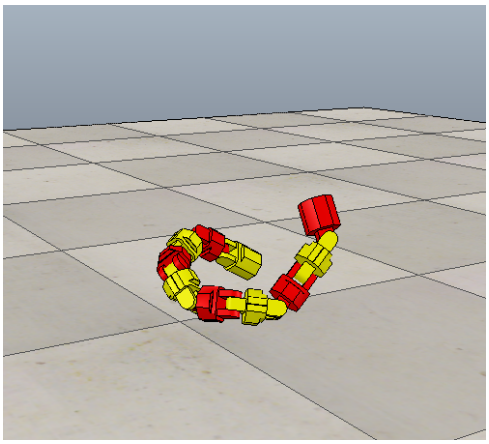
No actual trial was performed for this gait to prevent the robot from being damaged when clamping and rolling on a pole. Therefore, the gait was tested using the simulator. The increase in A alone may not be adequate for the robot to clamp around the pole, so it might be necessary to set $\psi > 0$. When $\psi > 0$, A can be increased to more than 68° depending on the magnitude of ψ. As ψ increases, the body of the snake robot starts to coil. The larger the ψ, the smaller the diameter and the longer the length of the coil as presented in Figure 4.25. It is important to note that a smaller A has a smaller maximum ψ for the shape of the snake robot to be practical for pole climbing. If ψ were too large, the diameter of the coil would be too small. A and ψ are at the optimum when a robot could clamp and roll effectively on a pole. They can be optimised using the annealed chain fitting method proposed by Hatton and Choset [49]. Furthermore, the complaint control proposed by Rollinson and Choses [43] enables a snake robot to adapt to the change of the pole size while climbing up the pole without compliance and torque sensing in the joints.



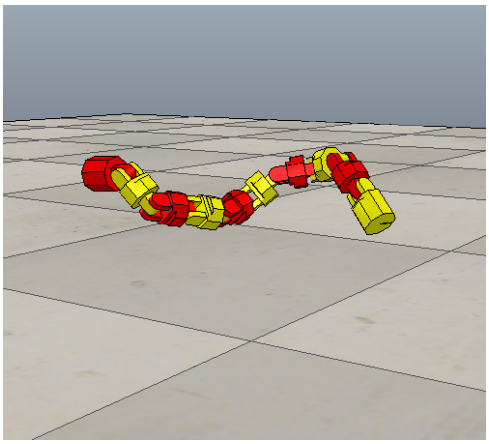
(a) $A_p = 30$, $\psi_p = 144$, and $O_y = 10$.



(b) $A_p = 30$, $\psi_p = 144$, and $O_y = 30$



(c) $A_p = 30$, $\psi_p = 144$, and $O_y = 60$



(d) $A_p = 30$, $\psi_p = 144$, and $O_y = -60$

Figure 4.25: The diameter and length of the coil shape with different magnitudes and signs of A and ψ.

4.6 Summary

The PID gains of the servo were optimised to minimise the errors in the joint angles of the robot. It was found that the servo requires P and D gains only to meet most of the requirements of the step-response characteristics. It was found that the optimal PD control failed to meet the settling time, but this was not an issue as the joint would have reached the target angle before it settled. However, it was found that the joint angles had larger errors when there was the presence of load that was due to the lack of torque of the servo. The lack of torque of servos had an impact on the performance of the joints while performing gaits. It was found that the errors in the joint angles for the sidewinding gait were the largest followed by the linear progression, the rotating, the rolling and the lateral undulation gaits. Unlike other gaits, the area of the ground contacts during the sidewinding gait resulted in the large errors. On the other hand, the robot was slightly tilted during the rolling and the rotating gait, and this resulted in even distribution of load among the pitch and the yaw modules. Load exerted is minimal to the yaw modules as indicated by the errors in the joint angles of the lateral undulation gait.

The conditions of optimal gait parameters for various gaits are as presented in Table 4.5. Parameters without subscripts mean that the pitch and yaw parameters are equal, and all the parameters not in the table are assumed to be zero, except for the angular speed ω which should be set below 6.28 rad/s. It is important to ensure that the modular snake robot remains in balance during the linear progression gait by setting $A < 60$. The body of the snake robot curls into arc shape during the turning and the lateral rolling gaits, meaning that the head and the tail modules would collide if A or O_y are set too high. It is found that they should not be set greater than 68° for the robot while performing the gaits.

Table 4.5: The conditions of optimal gait parameters for the modular snake robot.

Gait	Conditions of Optimal Parameters
Linear Progression	$0° < A_p < 60°$, $\psi_p > 108°$
Turning	$0° < A_p < 60°$, $\psi_p > 108°$, $O_y < \|68°\|$
Sidewinding	$A_p >= 5°$, $A_y >= A_p$, $\psi > 120°$, $\psi_{py} \neq 0.5\psi$
Rotating	$A_p >= 5°$, $A_p = 0.2A_y$, $\psi_p > 90°$, $\frac{\psi_p}{\psi_y} = 0.7$
Lateral Rolling	$0° < A <= 68°$, $\psi = 0°$, $\psi_{py} = 90°$
Pole Climbing	$A \neq 0°$, $\psi > 0°$, $\psi_{py} = 90°$

The friction affects the rotating gait significantly, so it is important to ensure that $A_p >= 5$. As for the pole climbing gait, trials on the actual robot were not performed; instead, only the robot in the simulator was used. The robot needs to clamp around a pole first before it starts climbing on it. For the robot to clamp around the pole, it is important to set the robot into a shape that is effective for the robot to roll on the pole.

Chapter 5

Motion Capture with a Calibrated Camera

5.1 Introduction

The motion of the snake robot can be divided into 'internal' and 'external' motions. The internal motion is the motion of the snake's joints whereas the external motion is the motion resulting from the internal motion. The common tracking methods used to capture the motion of a robot are optical-passive marker, optical-active marker, inertial, magnetic, mechanical, and acoustic [92]. However, not all of the tracking methods are applicable for external motion. For instance, the inertial sensors applied by Rollison et al. [42] were only limited to providing feedback for the relative joint position of the snake robot.

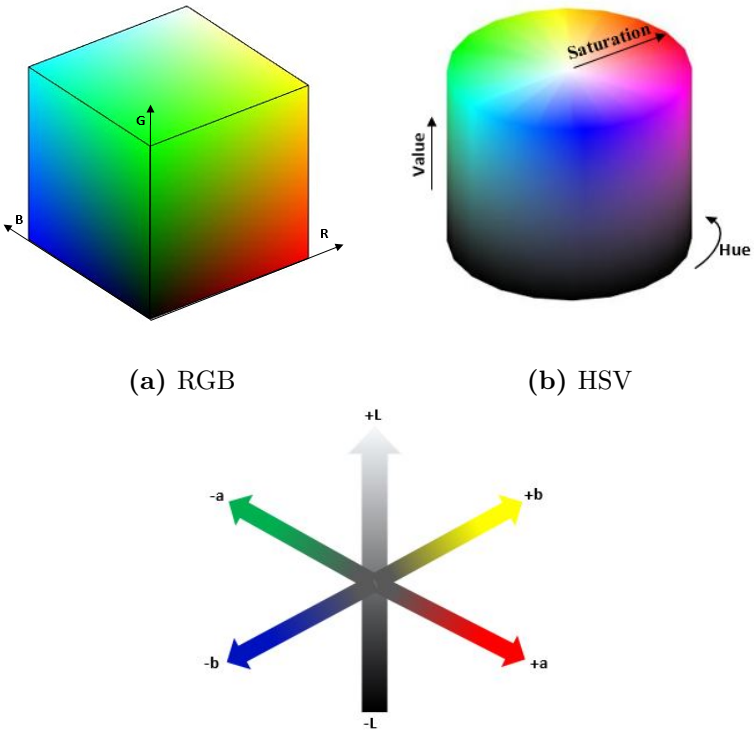
The most practical method that is used to capture the external motion of the snake robot is the optical-passive marker method as implemented by Xiao et al. [17] and Sato et al. [76]. Though using more than two cameras as proposed by Xiao et al. [17] could increase the accuracy and add another dimension to the motion capture, a single camera as proposed by Sato et al. [76] would be adequate for this research. Since the main focus of this thesis is to study the locomotion of a modular snake robot on a planar surface, the information of the depth captured from a stereo setup is unnecessary, and the frames captured by the camera are reprojected into a plane in 3D world coordinates.

This chapter covers the approach taken to perform the external motion capture using optical-passive markers. This approach involved camera calibration and colour segmentation. The results of the method were analysed to determine the measurement error of the motion capture. The measurement error of the calibrated camera might affect the results of the analysis performed on the motion trajectories in the future chapter.

5.2 Background

5.2.1 Colour Segmentation

Human perception of colour is a combination of primary colours which are red, green and blue. This representation known as RGB space is widely applied in computer graphics [93]. The RGB space can be represented as a cube graph as shown in Figure 5.1a.



(a) RGB **(b)** HSV

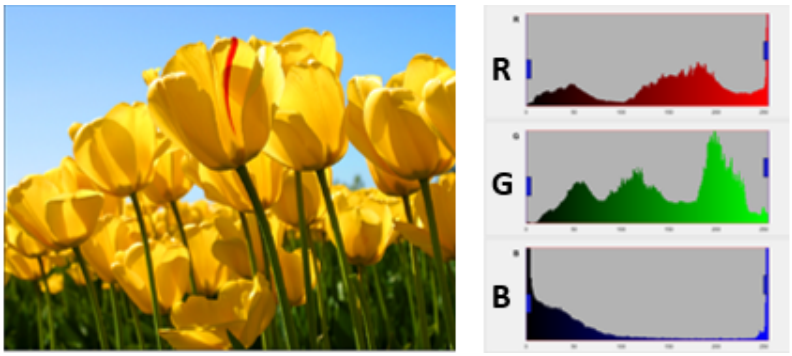
(c) LAB

Figure 5.1: RGB, HSV and LAB colour spaces.

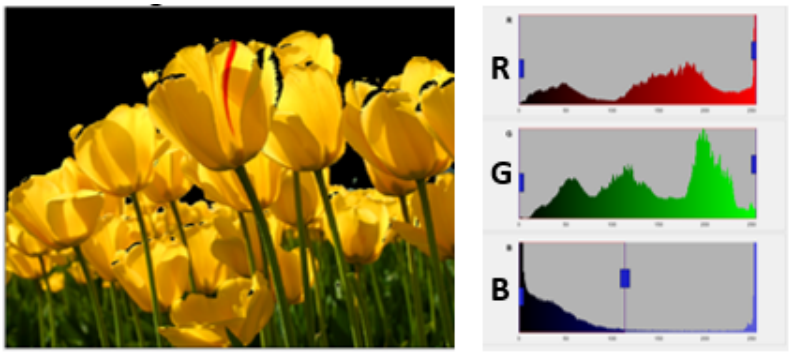
Using linear or nonlinear transformations, other colour spaces can be derived from RGB space. Linear colour spaces such as YIQ and YUV are used to encode colour information in television broadcasting. Due to the high correlation among the R, G, and B components, RGB is not a good choice for colour segmentation [93]. Due to this reason, colour spaces derived from nonlinear transformations such as HSV and CIE (also known as LAB) are common for colour segmentation [94]. HSV stands for hue, saturation, and value, and it can be modelled as a cylindrical surface as presented in Figure 5.1b. Figure 5.1c shows the model of CIE which consists of L, a and b components.

In colour segmentation, there are several methods: histogram threshold, colour space clustering, region-based approaches, edge detection, etc. The most common method used

in colour segmentation is histogram threshold. Histogram threshold is performed in each channel of the RGB space to distinguish the desired colour from the rest as demonstrated in Figure 5.2 [93]. On the other hand, HSV or CIE space only requires a slight adjustment to one or two channels to change the lightness and colourfulness of the desired colour. According to Sural et al. [95], HSV was able to classify colours with various intensities more accurately than RGB. It was also found that HSV performed better than CIE in colour segmentation for noisy colour images. However, the studies conducted by Khatab et al. [96] and Wang et al. [97] presented contrary results. Thus, it would be best to test some colour spaces to obtain the best results for this application. The choice of a colour space will depend on the segmentation method and the application of the colour segmentation.



(a) Before applying histogram threshold



(b) After applying histogram threshold

Figure 5.2: The outcome of an RGB image being applied with histogram threshold.

5.2.2 Camera Calibration

A camera converts an object into an image based on its perspective view of the object. In an ideal camera model, it is assumed that the camera has no lens [98]. With this assumption, the model of the camera is more simplified and the distortion caused by the

lens can be neglected by estimating the conversion parameters of the camera. In the camera model, the conversion of an object from the world to an image occurs in two steps as presented in Figure 5.3. First, the 3D world points are converted to 3D camera coordinates using *extrinsic parameters*. The image is then formed when the 3D camera coordinates are mapped into 2D camera coordinates using *intrinsic parameters*. Extrinsic parameters describe the pose of the camera in the 3D world coordinates whereas intrinsic parameters are the focal length, optical centre and skew coefficient of the camera. This model can be expressed as follows:

$$sp = C \begin{bmatrix} R & t \end{bmatrix} P \tag{5.1}$$

where s is the scale factor, $p = \begin{bmatrix} u & v & 1 \end{bmatrix}$ is a point in the 2D camera coordinates, $P = \begin{bmatrix} x & y & z & 1 \end{bmatrix}$ is a point in the 3D world coordinates, R is rotation, t is translation.

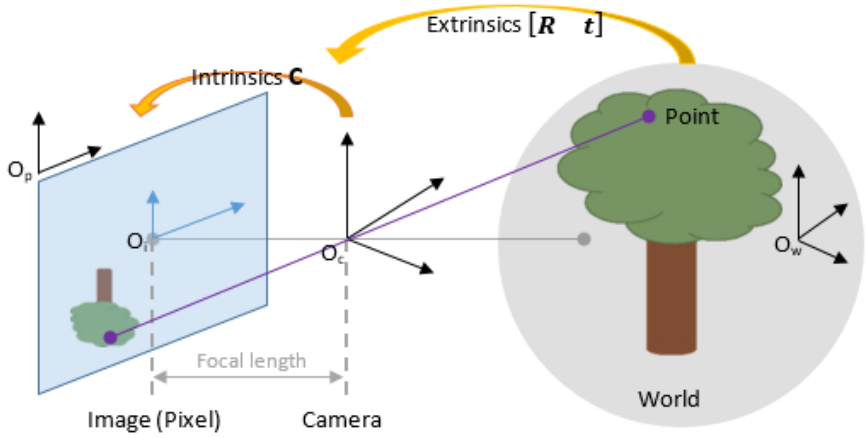


Figure 5.3: The conversion of an object from the world coordinates to an image.

C is the camera matrix which can be expressed as follows:

$$C = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

where f_x and f_y are the focal length (in pixels) in the x-axis and y-axis, γ is the skew coefficient, and c_x and c_y (in pixels) are the coordinates of the optical center. Using the technique proposed by Zhang [99], the intrinsic and extrinsic parameters can be obtained without the prior knowledge of the rotation and the translation of the camera. This requires at least two images of a planar pattern being taken at different orientations. Though the camera model has no lens, the radial distortion can still be estimated by another technique proposed by Zhang [99].

5.3 Camera Calibration

5.3.1 Methods

The camera used for this experiment was a Logitech HD Webcam C525 capturing images at 1280 × 720 resolution pixels at 25 frames per second, with auto lens zooming feature being disabled. Since this research involved capturing the motion of the snake robot from videos, it was necessary to calibrate the camera once to obtain the extrinsic and intrinsic parameters so that the frames from the video could be reprojected into a plane in 3D world coordinates. 20 images of a chess board were captured from various angles and distances. The chess board was made up of 7 rows and 10 columns of 30 mm boxes. In one of the images, the chess board was placed parallel to the ground and aligned in the same orientation as the frame of the camera. This image was used as reference for the frames to be reprojected into a plane in the simulated 3D world coordinates.

The intrinsic and extrinsic parameters of the camera were obtained using the Camera Calibration Toolbox on MATLAB (R2016A, The Mathworks, Natick, MA) [100]. The parameters were used to reproject the snake robot in the image to a plane in the simulated 3D world coordinates. The average reprojection error obtained from the toolbox was used to determine the quality of the calibration.

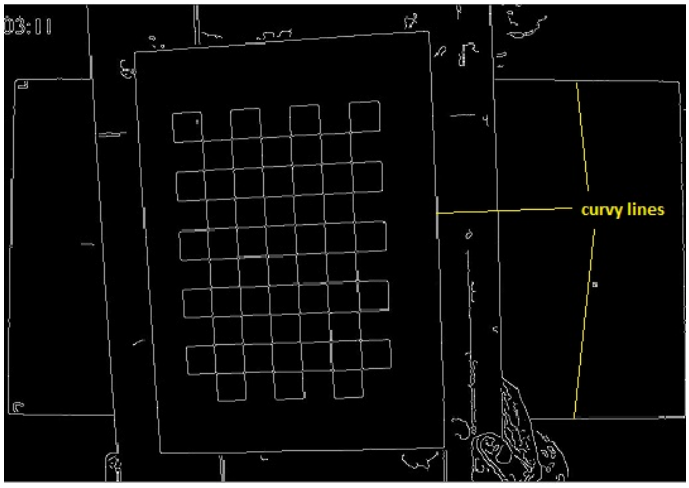
To improve the camera calibration quality, poor image samples with large errors were discarded and replaced with better image samples and the camera was re-calibrated [101]. Removing the distortion in the frames due to lens distortion was necessary to improve the accuracy of reprojection of 2D plane to 3D space. The distortion parameters obtained from the camera calibration toolbox were used to rectify the frames of the video before performing motion capture.

5.3.2 Results and Discussion

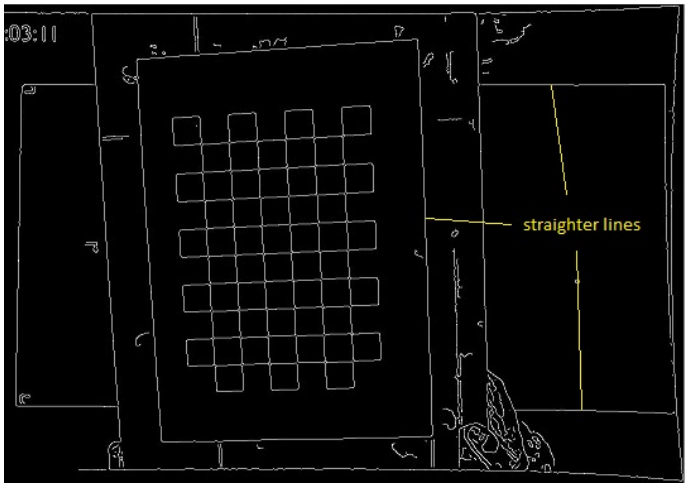
The intrinsic, extrinsic, and distortion camera parameters obtained from the camera calibration toolbox are presented in Table 5.1. Using these parameters, the positive radial ('barrel') distortion was removed from the image captured by the camera. Since the distortion was not apparent, "Canny" edge detection proposed by Canny [102] was used to make the distortion more apparent as shown in Figure 5.4. As shown in Figure 5.4b and 5.4a, the rectified image displayed straighter edges compared to that of the distorted image.

Table 5.1: The intrinsic and extrinsic parameters of the camera used in the experiment.

Parameter	Results
Focal length	$[1640, 1640]$ pixels
Principal point	$[962, 497]$ pixels
Rotation	$\begin{bmatrix} 0.0012 & -0.9996 & -0.9996 \\ 0.9998 & 0.0006 & -0.0218 \\ 0.0217 & 0.0274 & 0.9994 \end{bmatrix}$
Translation	$[-0.739, 0.0816, 2.6146]$ pixels
Radial distortion	$[-0.0285, -0.0415]$ pixels



(a) Distorted image



(b) No distortion

Figure 5.4: The lines appeared to be straighter in the image with no distortion.

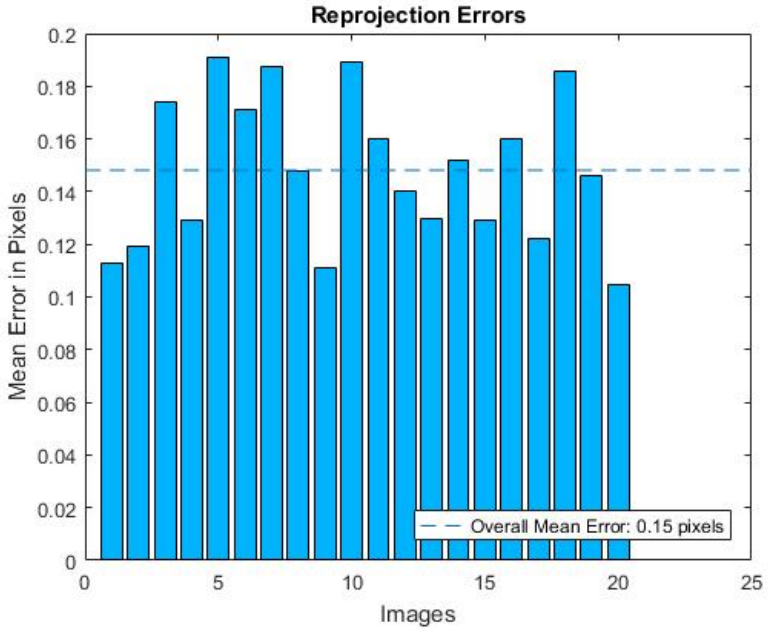


Figure 5.5: Reprojection error from 2D plane into 3D world coordinates for each frame used for the camera calibration.

The toolbox also determined the reprojection errors from the 2D plane into the 3D space of each image sample used in the camera calibration. The reprojection error for each sample image used in the camera calibration is as shown in Figure 5.5. After several attempts of improving the mean reprojection error, the best mean reprojection error obtained was 0.15 pixels. This error gave the general idea on how the camera alone would affect the accuracy of the measurement of the calibrated camera but it did not provide the exact error in terms of the 3D space. The reprojection error calculated from the actual measurement of an object in the image would be more precise.

5.4 Colour Segmentation

5.4.1 Methods

To determine the position and orientation of the snake robot, a pair of markers were placed at both ends of each part of the snake: the head, the middle and the tail modules as shown in Figure 5.6. Colour segmentation was performed to detect the markers. The three colours (red, green, and blue) were used because their hue values are further apart in HSV space and LAB space.

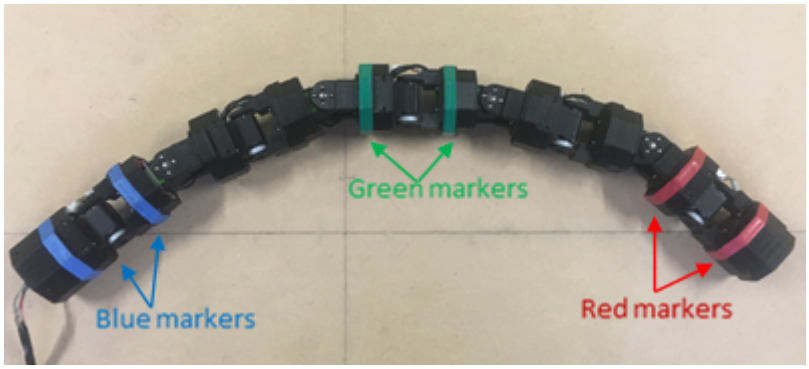
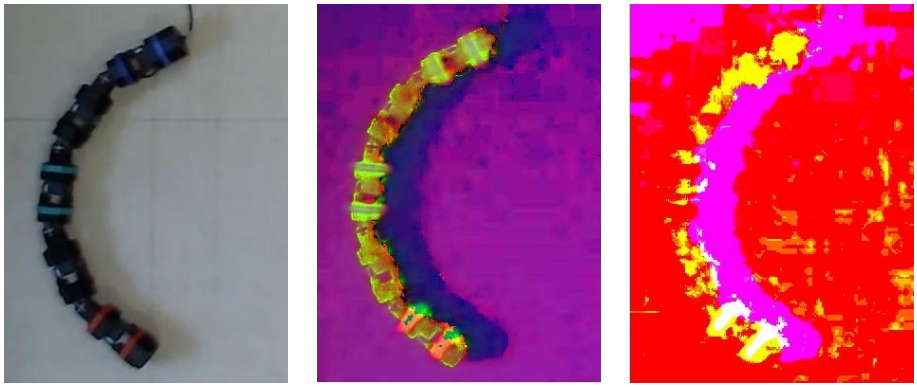


Figure 5.6: The head, middle and tail modules of the robot were respectively marked with red, green, and blue PVC tapes.

To perform colour segmentation in either colour space, a frame was first converted to the format of the desired colour space as shown in Figure 5.7. Colour threshold was performed on the converted frames in RGB, HSV, and LAB colour spaces resulting in binary images. For instance, colour threshold in RGB for detecting markers can be expressed as in Equation 5.3.

$$\begin{aligned} T(R,G,B) = R_{min} < I(R,:,:) <= R_{max} \ \text{AND} \\ G_{min} < I(:,G,:) <= G_{max} \ \text{AND} \\ B_{min} < I(:,:,B) <= B_{max} \end{aligned} \tag{5.3}$$



(a) RGB **(b)** HSV **(c)** LAB

Figure 5.7: The HSV and LAB images presented in RGB format.

R, G, and B are the three colour channels in RGB space. Colour threshold in HSV and LAB spaces was performed using the same equation by respectively replacing the R, G, and B channels with H, S, and V channels and L, a, and b channels. Table 5.2, 5.3, and 5.4 present the range of each channel used in the colour threshold to detect red, green, and blue markers.

Six videos of rolling motion trials (600 total frames) were used as samples to test the performance of the image segmentation using RGB, HSV, and LAB spaces. The performance of the detection was determined by its failure rate in detecting the markers. The failure rate was categorised into false positive and false negative. The false positive occurred when non-markers were detected as markers, whereas false negative occurred when markers were not detected.

Table 5.2: The range of three channels to detect red, green and blue makers in RGB space.

Marker Colour	R_{min}, R_{max}	G_{min}, G_{max}	B_{min}, B_{max}
Red	60, 255	0, 50	0, 70
Green	0, 40	60, 255	0, 105
Blue	0, 50	0, 80	80, 255

Table 5.3: The range of three channels to detect red, green and blue makers in HSV space.

Marker Colour	H_{min}, H_{max}	S_{min}, S_{max}	V_{min}, V_{max}
Red	$325°, 7°$	0.30, 1.00	0.20, 1.00
Green	$160°, 200°$	0.32, 1.00	0.30 ,1.00
Blue	$223°, 292°$	0.58, 1.00	0.22, 1.00

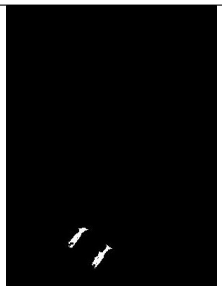
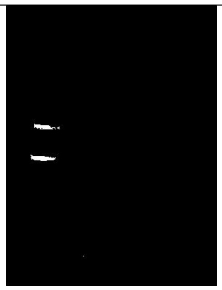
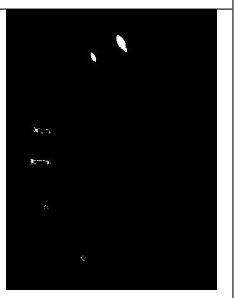
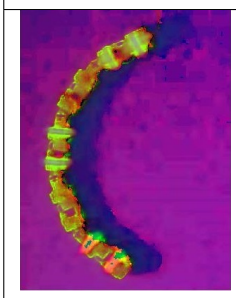
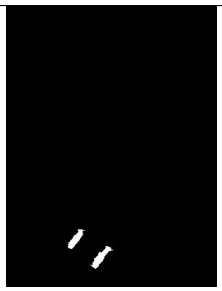
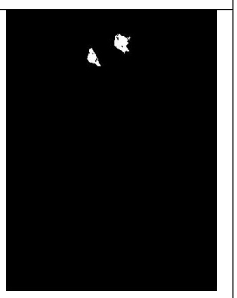
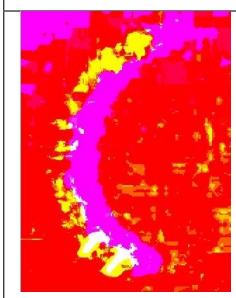
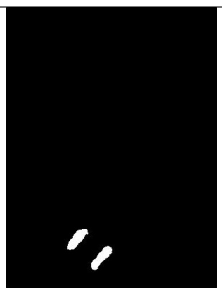
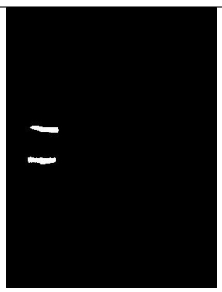
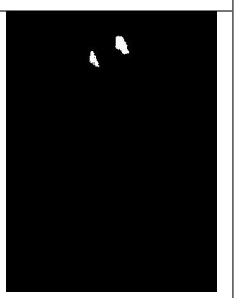
Table 5.4: The range of three channels to detect red, green and blue makers in LAB space.

Marker Colour	L_{min}, L_{max}	a_{min}, a_{max}	b_{min}, b_{max}
Red	9, 44	5, 25	-14, 27
Green	0, 100	-19, -7	-31, -8
Blue	0, 70	8, 27	-37, -13

5.4.2 Results and Discussion

The outcomes of the colour segmentation using the threshold ranges as listed in Table 5.2, 5.3, and 5.4 are presented in Table 5.5. The performance of the colour segmentation using RGB, HSV, and LAB spaces is presented in Table 5.6. Based on Table 5.6, RGB space gave the poorest results compared to other colour spaces. Unlike HSV and LAB, RGB gave false negative outcomes. This is because it was often difficult to distinguish colour using RGB especially with the presence of light reflection. The light reflection caused the green markers and the body of the snake to look slightly bluish.

Table 5.5: The outcomes of colour segmentation for detecting red, blue and green markers in RGB, HSV and LAB colour spaces.

Colour Space	Red	Green	Blue
			
			
			

False positive errors were obtained at a significantly lower rate with HSV and LAB spaces. The colour segmentation with LAB space performed better than HSV space. However, the time taken to convert an RGB image format to LAB format was twice that to HSV when using MATLAB. To detect the markers in real-time, HSV would be the best choice for detecting the markers.

Table 5.6: Performance of colour segmentation using RGB, HSV, and LAB colour spaces.

Failure Type	RGB (%)	HSV (%)	LAB (%)
False Positive	20.3	12.7	9.7
False Negative	3.1	0	0

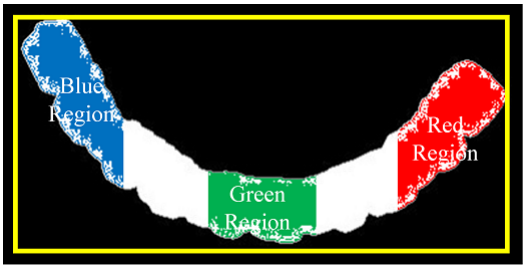
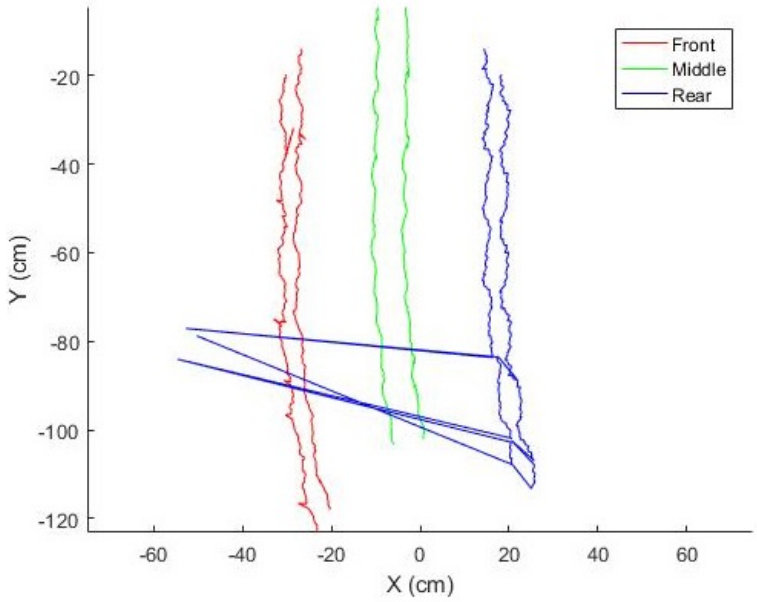
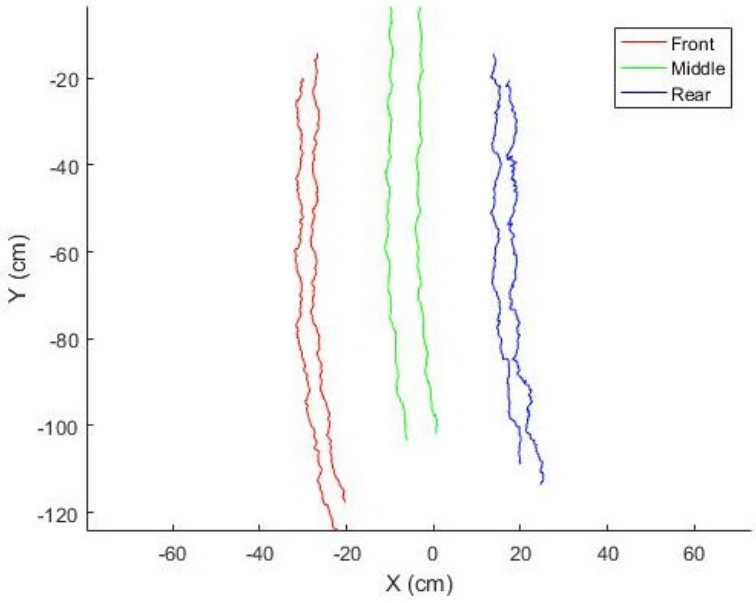


Figure 5.8: The regions which the red, green and blue markers would likely to be detected.



(a) Before error removal



(b) After error removal

Figure 5.9: The quality of the motion trajectory before and after removing the errors ('Front', 'Middle', and 'Rear' are the head, the middle and the tail modules).

To ensure all errors from the marker detection were eliminated, the smoothness of the captured 'external' motion trajectory was examined. Since the failure rate of marker detection with colour segmentation alone was still high, regional classification of the markers, as shown in Figure 5.8, was implemented to significantly reduce the errors. With regional classification, any coloured pixel detected outside of the colour regions would be removed. Regional classification was obtained by discarding the contrasting background in the frame, leaving only the silhouette of the snake robot. The positions of the markers in the initial frame of a trial served as a reference for the regional classification. With fewer errors in the captured motion trajectory, it was easier to manually eliminate the errors by individually examining the frames which failed in the colour segmentation. To eliminate the errors, the colour threshold was slightly adjusted to detect the markers in the frames more accurately. Removing all errors made the captured motion trajectory appear much smoother as presented in Figure 5.9.

5.5 Reprojection Error

5.5.1 Methods

The actual reprojection error was determined by comparing the measured markers' positions with their actual positions, similar to the experiment presented on the MATLAB website [103]. The surface board was marked with seven lines at an equal distance of 29 cm apart from one another as shown in Figure 5.10.

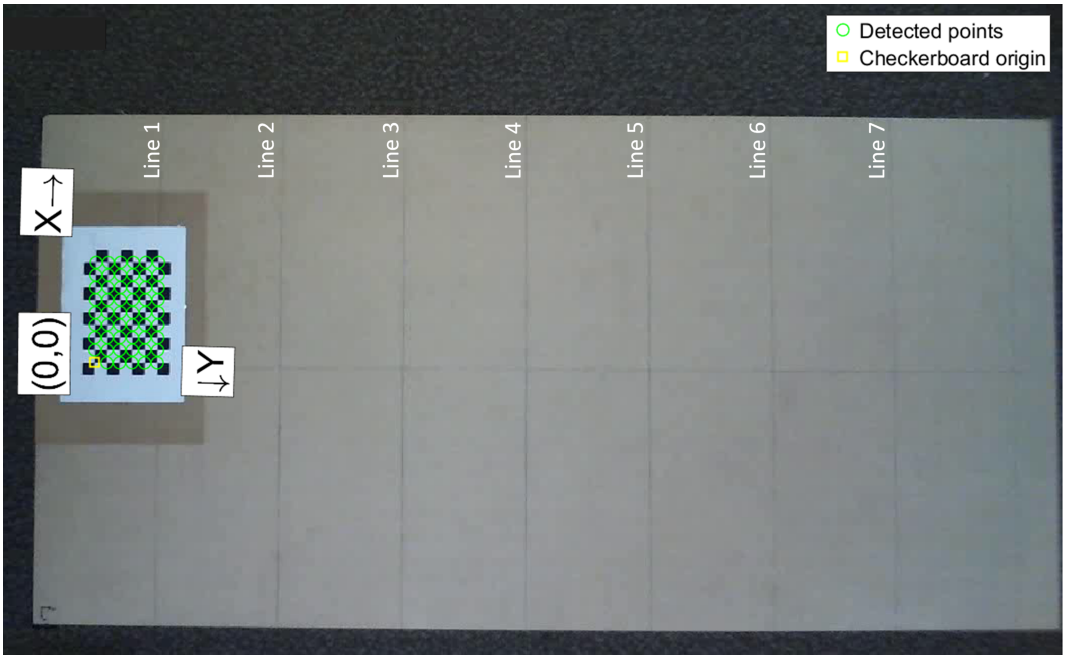


Figure 5.10: The surface board used to determine the overall reprojection error.

Before the reprojection errors can be determined, the snake robot was straightened when it was placed at each of the lines. The robot was placed in a way that the red markers are at the positive side and the blue markers are at the negative side of the coordinates, as defined in Figure 5.10. The frames in which the robot was placed at each line were captured, and the motion capture reprojected the positions of the markers at all the lines. The distances between the markers from one line to another line were used to determine the reprojection error in the y-axis. Figure 5.11 shows that the green and the blue markers, and the blue and the red markers are 27.2 cm apart when the robot is fully straightened. These distances were used to determine the reprojection error in the x-axis.

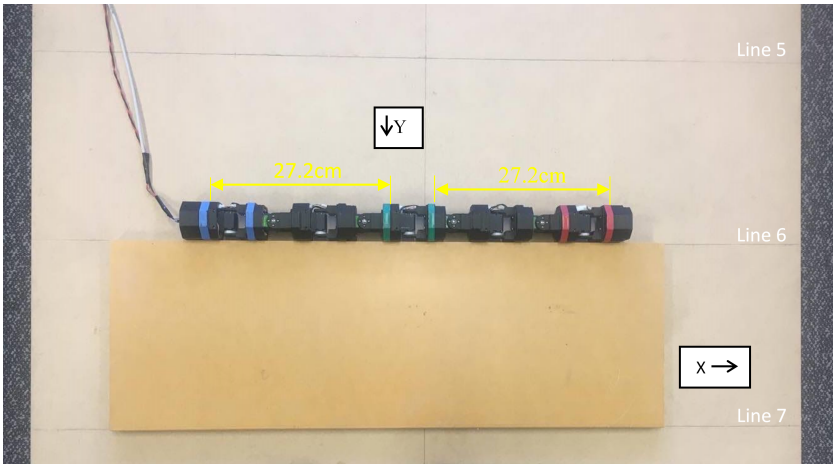


Figure 5.11: The distances between the two different colour markers.

5.5.2 Results and Discussion

Figure 5.12 shows the positions of a pair of red, blue and green markers at each line of the surface board in a generated 3D virtual world. The coordinates of the three different colour markers measured by the motion capture and their midpoint coordinates are presented in Tables B.1-B.2 (see Appendix B). The 18 measured distances of two lines, the 14 measured distances of two different colour markers, and the errors in the measurements are presented in Table B.3 (see Appendix B). Based on Table B.3, the average reprojection errors in the x-axis and the y-axis are +1.5% and +1.2% respectively. These errors are difficult to minimise as they were mainly contributed by the outcomes of the colour segmentation.

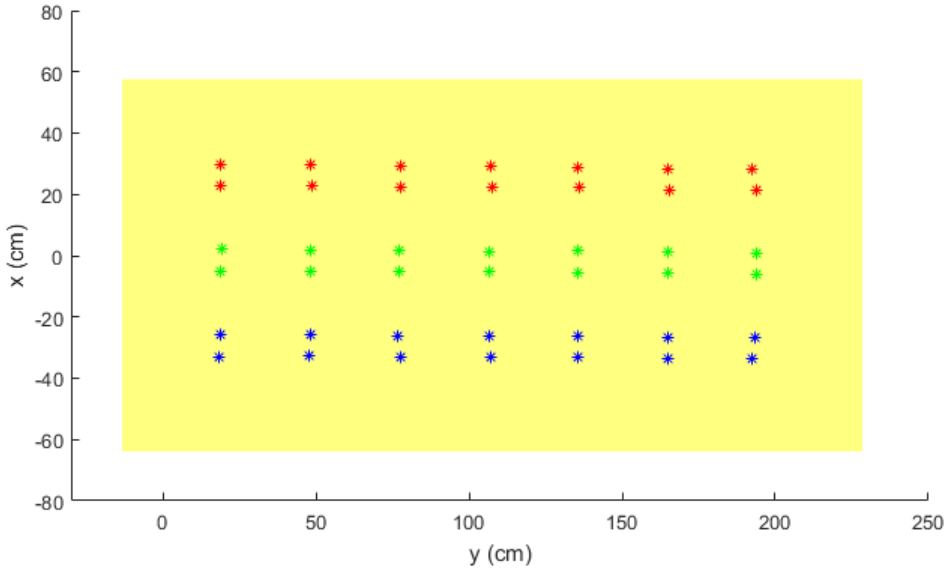


Figure 5.12: The 3D virtual world generated using the information gathered from the reprojection.

5.6 Summary

To capture the external motion of the modular snake robot, there were two main steps involved: camera calibration and colour segmentation. The intrinsic and extrinsic parameters obtained from the camera calibration were used to reproject an object in the 2D camera coordinates to a plane in the simulated 3D world coordinates generated by the MATLAB. The performance of the camera calibration was measured using the average reprojection error from the MATLAB Toolbox and the best calibration outcome was obtained when the error was minimised to 0.15 pixels. Six recorded trials of rolling gait were used to test the performance of the colour segmentation in RGB, HSV and LAB spaces. Red, green and blue markers were used to mark the head, the middle and the tail of the snake robot. RGB gave the worst performance as compared to other colour spaces and LAB performed slightly better than HSV. To minimise the failure rate, the frames with failures were examined manually to determine the snake robot's position. Since colour segmentation contributes to the reprojection error, an experiment which compared the actual distance and the reprojected distance was conducted, and it was found that the actual reprojection errors in the x-axis and the y-axis are +1.5% and +1.2% respectively. These errors would be considered as the uncertainty for the motion analysis conducted in the future chapter.

Chapter 6

Motion Trajectory Analysis

6.1 Introduction

Predicting the motion trajectories generated by different sets of gait parameters is important for human operators to control snake robots and perform path-planning. The virtual chassis enables human operators to visualise the locomotion of snake robots based on their overall shapes. The joints of snake robots can also be easily tracked using the method.

To predict the motion trajectories of a modular snake robot, Enner et al. [34] proposed a simplified kinematic motion model based on the virtual chassis to estimate the external motion of the robots. A further study has been conducted in conjunction with this thesis. This study modifies the model to consider the centre of mass of a snake robot. This modification has slightly improved the accuracy of predicting the motion trajectory for the rolling gait [104].

The purpose of this chapter is to determine the accuracy of motion trajectories predicted by the modified 3D motion model and the developed 2D motion model, which are based on the virtual chassis. The chapter presents the inconsistency in the recorded motion trajectories for linear progression, turning, rolling, rotating and sidewinding gaits, which were generated using different sets of gait parameters. The chapter also presents the average optimal ground parameters, τ and δ, which were obtained by training models with the recorded motion trajectories. The accuracy of the models for the gaits are then presented in terms of errors of the travelled distance, the direction and the orientation of the snake robot.

6.2 Virtual Chassis

Snake robots have non-rigid bodies which can be formed into various shapes. Tracking the position of each joint of the snake robot can be complicated and is computationally expensive so it is necessary to define the body frame of the robot that separates its internal motion from the world frame. The body frame can be defined using the virtual chassis proposed by Rollison [33] to align the internal shape of the robot to its principal moment of inertia. The virtual chassis enables the visualisation of the changes in the internal shape of a snake robot, as illustrated in Figure 6.1. Without the virtual chassis, it would be difficult to track the changes in the internal shape of the robot..

Before the virtual chassis of a snake robot can be determined, the robot needs to be inside a fixed-world frame that is computed using homogeneous transformation matrices [105]. Since a pitch-and-yaw snake robot was used as the robotic platform in this study, each joint of the robot rotates about the y-axis and the z-axis relative to the body frame of each of its modules. The first module, the 'head', is plotted at the origin followed by the next consecutive modules. This plot generates a fixed-world frame as shown in Figure 6.1b. To determine the virtual chassis of the robot, a data matrix, P, consisting of the position of each module relative to the centre of mass of the robot, $\bar{p} = [\bar{x}, \bar{y}, \bar{z}]^T$, is firstly constructed as presented in Equation 6.1. Singular Value Decomposition (SVD) is then applied to the matrix, P, to obtain the rotation matrix, V, as shown in Equation 6.2.

$$P = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ x_2 - \bar{x} & y_2 - \bar{y} & z_2 - \bar{z} \\ \vdots & \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} & z_n - \bar{z} \end{bmatrix} \tag{6.1}$$

$$USV^T = P \tag{6.2}$$

Since the sign convention of rotation matrix, V, is ambiguous, it is important to define its sign convention at the initial time, $t = 0$ [33]. It can be defined using the Equation 6.3 where $v1$, $v2$ and $v3$ are the first, the second and the third column vectors (3×1 matrices) of the rotation matrix, V, and t is the time period. At $t = 0$, V needs to be aligned to the world frame using dot product between the column vectors of V, and the column vectors of an identity matrix, I, as shown in Equation 6.4 where $s1$, $s2$ and $s3$ are the signs to define the sign convection of the matrix [106].

$$V_t = \begin{bmatrix} v_{1,t} & v_{2,t} & v_{3,t} \end{bmatrix} \tag{6.3}$$

$$
\begin{aligned}
s_1 &= v_{1,0} \bullet \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
s_2 &= v_{2,0} \bullet \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\
s_3 &= v_{1,0} \times v_{2,0}
\end{aligned}
\tag{6.4}
$$

$$
\tilde{V}_0 = \begin{bmatrix} s_1 \cdot v_{1,0} & s_2 \cdot v_{2,0} & s_3 \cdot v_{3,0} \end{bmatrix}
\tag{6.5}
$$

The initial rotation matrix, $\tilde{V}_0$ can be expressed as shown in Equation 6.5. For the later time step, the column vectors of $\tilde{V}$ from the previous time step is used instead of the identity matrix, I, to ensure that sign flip does not occur across time step [106]. Thus, Equation 6.6 is used instead to find the sign of the column vectors.

$$
\begin{aligned}
s_1 &= v_{1,t} \bullet v_{1,t-1} \\
s_2 &= v_{2,t} \bullet v_{2,t-1} \\
s_3 &= v_{1,t} \times v_{2,t}
\end{aligned}
\tag{6.6}
$$

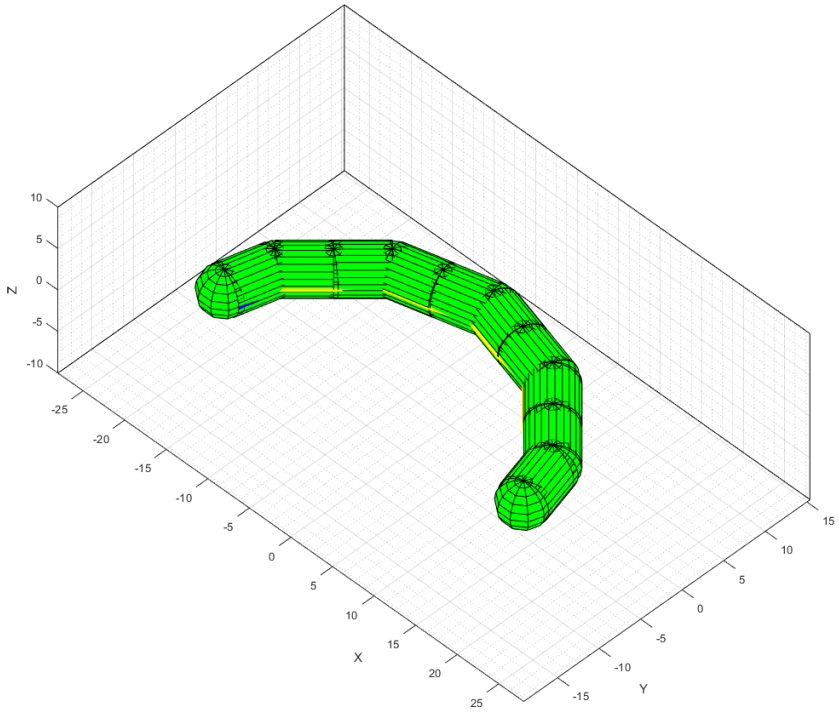
The homogeneous transformation matrix, T_v, is formed as shown in Equation 6.7.

$$
T_v = \begin{bmatrix} \tilde{V} & \bar{p} \\ 0 & 1 \end{bmatrix}
\tag{6.7}
$$

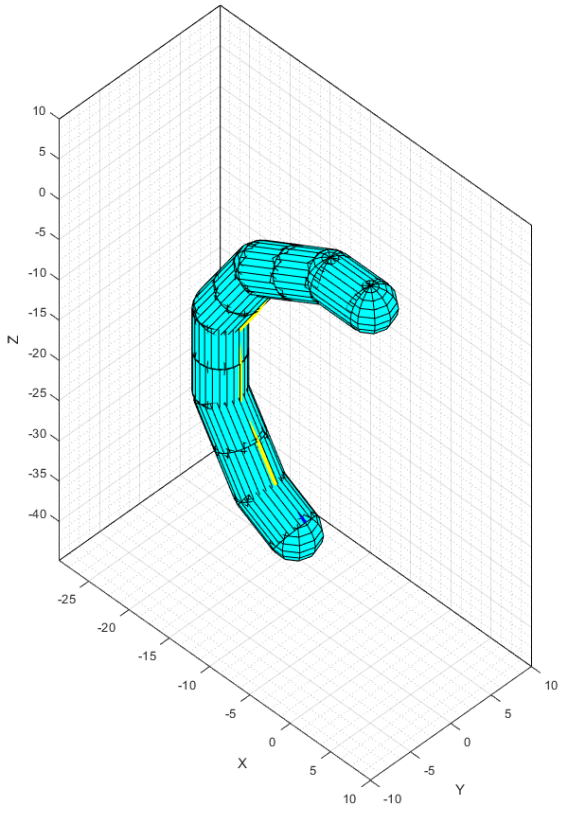
As shown in Equation 6.2, multiplying the position of each module with the inverse of the matrix, T^{-1}, gives the position of the module in the virtual chassis P_{vc} [33].

$$
P_{vc} = T_v^{-1} \cdot \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}
\tag{6.8}
$$

In the virtual chassis, the ground plane is always parallel to the flattest side of the snake's shape. This ground plane lies in the z axis perpendicular to x-axis and y-axis. It is found that this alignment makes it easy to model gaits when the yaw modules contribute the most in forming the shape of the robot because the x-y plane at the bottom of the snake can be assumed as the ground plane.



(a) Virtual chassis



(b) Fixed world frame

Figure 6.1: The internal shape of a snake robot while performing lateral rolling gait in the virtual chassis and the fixed-world frame.

However, problems arise when only pitch modules or both the yaw and pitch modules contribute equally in forming the shape of the robot. In this case, it is important to be aware that the robot's orientation may not be the same as the orientation of the actual snake robot. To adjust the snake's orientation, the homogeneous transformation matrix, T_v, is multiplied by a 4x4 rotational matrix as shown in Equation 6.2.

For instance, sidewinding gait can be set to perform slope climbing mode by setting the robot's body with equal magnitude of pitch and yaw waveform. In some gaits, the robot can be slightly imbalanced that causes its body to tilt periodically from side to side. The implementation of inertial sensors may help to re-orientate the snake to the same orientation as the snake robot. However, this effect of the tilt on the kinematic motion model will be neglected as the inertial sensors were not used in the study conducted by Enner et al. [34].

$$T_{v,f} = T_{v,i} \bullet \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\vartheta) & -\sin(\vartheta) & 0 \\ 0 & \sin(\vartheta) & \cos(\vartheta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.9}$$

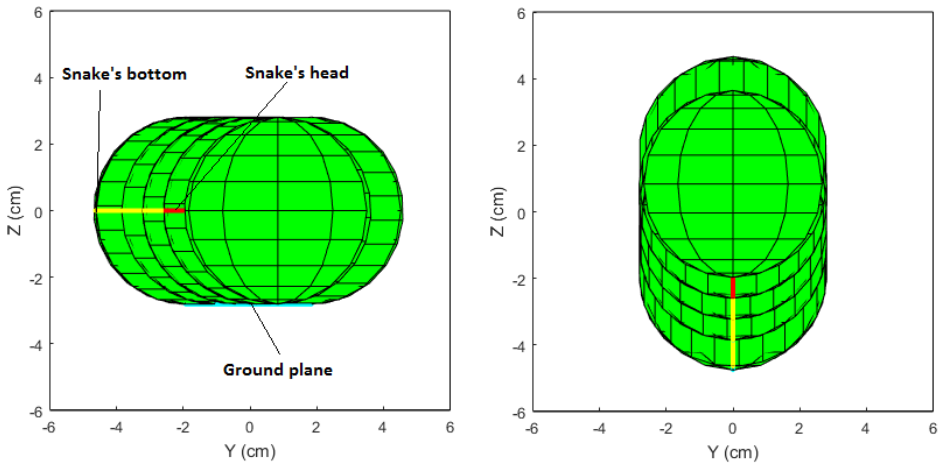
As for a linear progression gait, the shape of the snake robot, which is in vertical plane, failed to be computed into the virtual chassis. This problem can be resolved by replacing s_2 in Equation 6.4 as follows:

$$s_2 = v_{2,0} \bullet \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{6.10}$$

Though the issue is resolved, the orientation of the robot in the virtual chassis is incorrect, as shown in Figure 6.2a. The robot should be in the orientation as shown in Figure 6.2b. So, the robot needs to be rotated about 90° in the x-axis by multiplying the homogeneous transformation matrix, T, with a rotation transformation matrix as shown in Equation 6.2 where $\vartheta = -90°$

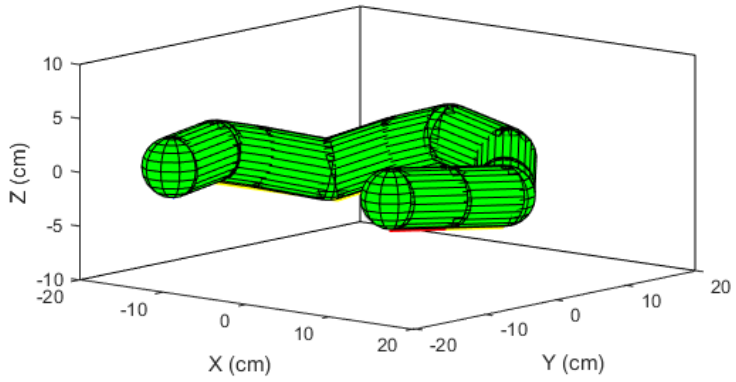
Figure 6.3b shows the robot is in an incorrect orientation as the tail module is supposed to be in the negative x-axis and both the tail and the head modules of the snake should be facing in the negative y-axis, as shown in Figure 6.3a. When computing a turning gait, the robot is in an incorrect orientation, as shown in Figure 6.3b, when $O_y > 45°$. To correct the orientation, the signs of the column vectors are negated as follows:

$$\begin{aligned} s_1 &= v_{1,0} \bullet \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} \\ s_2 &= v_{2,0} \bullet \begin{bmatrix} 0 & -1 & 0 \end{bmatrix} \end{aligned} \tag{6.11}$$

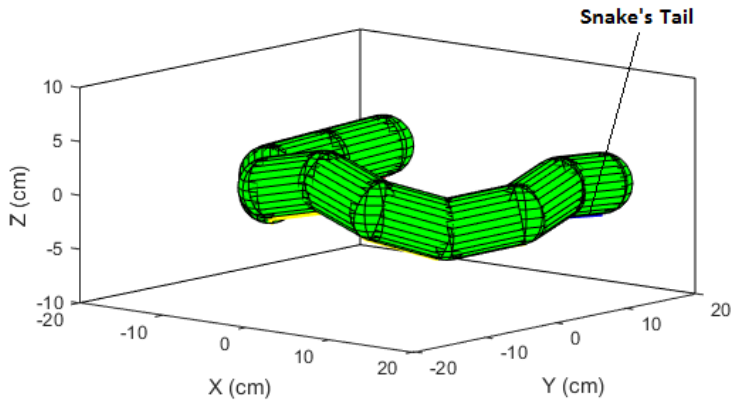


(a) Incorrect orientation **(b)** Actual orientation

Figure 6.2: The incorrect orientation of the robot in the virtual chassis and the actual orientation of the robot.



(a) Actual orientation

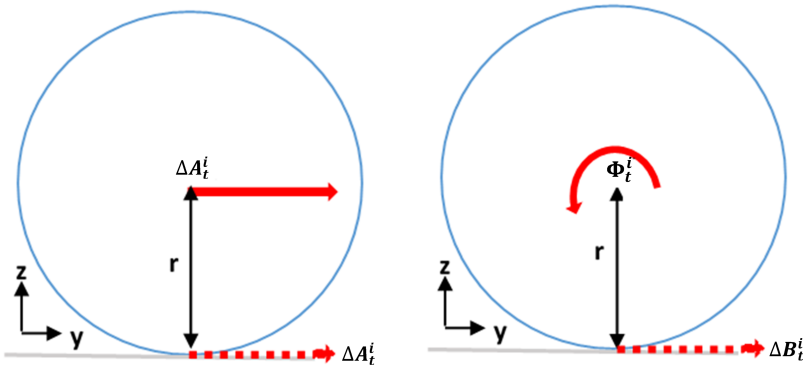


(b) Incorrect orientation

Figure 6.3: The orientation of the snake robot in the virtual chassis $O_y > 45°$ for turning gait.

6.3 Kinematic Motion Modelling for 3D Locomotion

The kinematic motion model proposed by Enner et al.[34] requires the snake robot to be inside the virtual chassis. As previously mentioned, the virtual chassis separates the internal motion of the robot from the world frame, enabling the position and orientation of the modules of the robot to be tracked in each time step. The contact between the lowest point of a module and the ground is assumed to roll similarly to a wheel as shown in Figure 6.4.



(a) Translation **(b)** Rotation

Figure 6.4: The translation and rotation motion.

Based on this assumption, the rolling motion induced by the translation, ${\Delta A_t}^i$, and the rotation, Φ_t^i, of each module can be expressed as in Equation 6.12 and 6.13.

$$ {\Delta A_t}^i = {A_t}^i - A_{t-1}^i \tag{6.12} $$

$$ \Phi_t^i = ({R^i}_{t-1})^{-1} {R_i}^t \tag{6.13} $$

$$ \Delta B_t^i = R_t^i \frac{\Phi_t^i r_t^t - (\Phi_t^i)^{-1} r_t^i}{2} \tag{6.14} $$

The translation displacement, ${\Delta A_t}^i$, and rotational displacement, ${\Delta B_t}^i$, of each module are added together as follows:

$$ {\Delta Q_t}^i = {\Delta A_t}^i + {\Delta B_t}^i \tag{6.15} $$

A ground contact model as in Equation 6.16 is applied in the motion model to weight the influence of the ground contact of each module on the external motion because not all modules are in the positions that contribute to the generation of the external motion [34].

$$ \gamma_t^i = \begin{cases} 1 - \frac{z_t^i - z_{min}}{\tau}, & \text{if } z_t^i - z_{min} < \tau \\ 0, & \text{otherwise} \end{cases} \tag{6.16} $$

The model is influenced by the ground parameters which are: the threshold parameter, τ, and the curvature parameter, δ. τ controls the level where a module is considered to be in contact with the ground, whereas δ controls the curve of the weighting function of the ground contact model. The effects of τ and δ are as shown in Figure 6.5. The ground model, γ_t^i, is used to find the weight of each module, which describes the contribution of the module to the external motion using Equation 6.17. w_t^i for each module is then normalised using Equation 6.18.

$$w_t^i = \frac{1 - \exp(-\delta\gamma_t^i)}{1 - \exp(-\delta)} \tag{6.17}$$

$$\hat{w}_t^i = \frac{w_t^i}{\sum_{i=1}^{n} w_t^i} \tag{6.18}$$

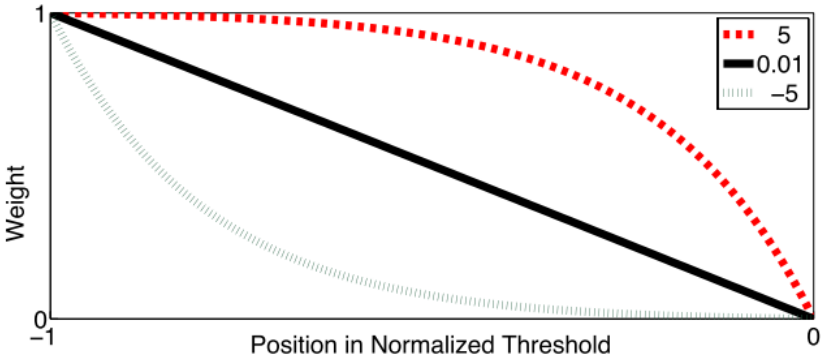


Figure 6.5: The characteristic of the weight, w_t^i, over three different curvature parameter, δ (retrieved from [34]).

The kinematic motion model calculates the translation and the rotation of the robot in the virtual chassis with respect to its centre of mass. The translation motion is calculated as follows:

$$\Delta M_t = -\sum_{i=1}^{n} \hat{w}_t^i \Delta Q_t^i \tag{6.19}$$

Before the rotational motion can be calculated, the moment vectors are obtained by performing cross product between each module's position in the virtual chassis as follows:

$$d_t^i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \times a_t^i \tag{6.20}$$

The vectors are then normalised as follows:

$$\bar{d}_t^i = \frac{d_t^i}{\|d_t^i\|} \tag{6.21}$$

The rotational motion is then calculated as follows:

$$\Delta\Phi_t = -\sum_{i=1}^{n} \hat{w}_t^i \frac{\Delta Q_t^i \cdot \bar{d}_t^i}{\|a_t^i\|} \tag{6.22}$$

Both translation motion, ΔM_t, and rotational motion, $\Delta\Phi_t$, are the approximated external motions based on the changes of the position and the orientation of the snake's modules in the virtual chassis. Since external motions are generated from the reaction forces which act in the opposite to the internal motions generated by the modules, the total translation and rotation in Equation 6.19 and 6.22 are flipped. The total translation and rotation are used to describe the changes of a snake robot's position and orientation relative to the world frame for a single time step as in Equation 6.23[34].

$$T = \begin{bmatrix} \cos\Phi_t & -\sin\Phi_t & 0 & \Delta M_t^x \\ \sin\Phi_t & \cos\Phi_t & 0 & \Delta M_t^y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.23}$$

The motion trajectories for a linear progression gait and a turning gait cannot be computed accurately. This is because, based on the translation of the 3D kinematic motion model, the robot is assumed to move back and forth as shown in Figure 6.6. Furthermore, the model only calculates the rotation motions which occur about the x-axis and the y-axis located at the centroid of the robot in the virtual chassis.

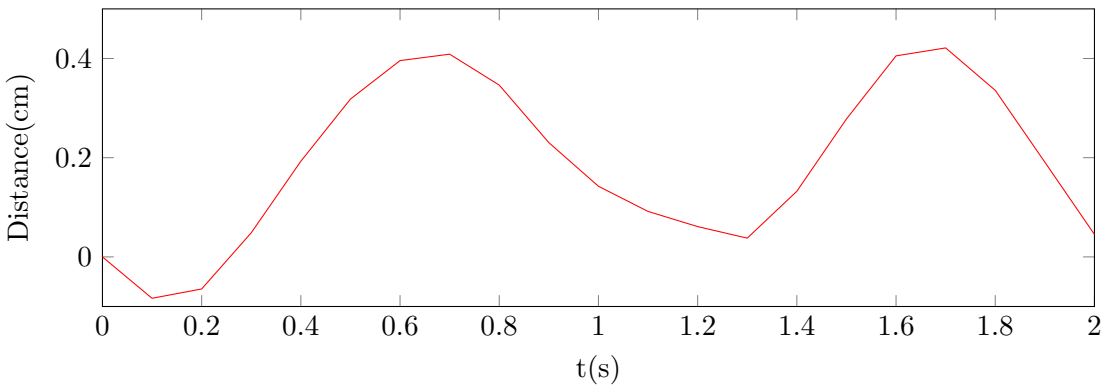
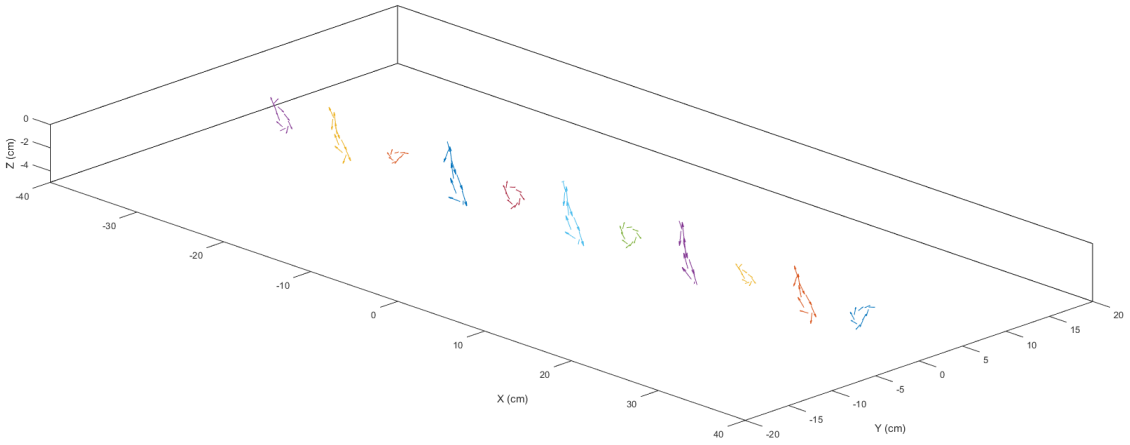


Figure 6.6: The computed distance travelled by the snake robot in a single cyclic of a linear progression gait using the 3D kinematic motion model.

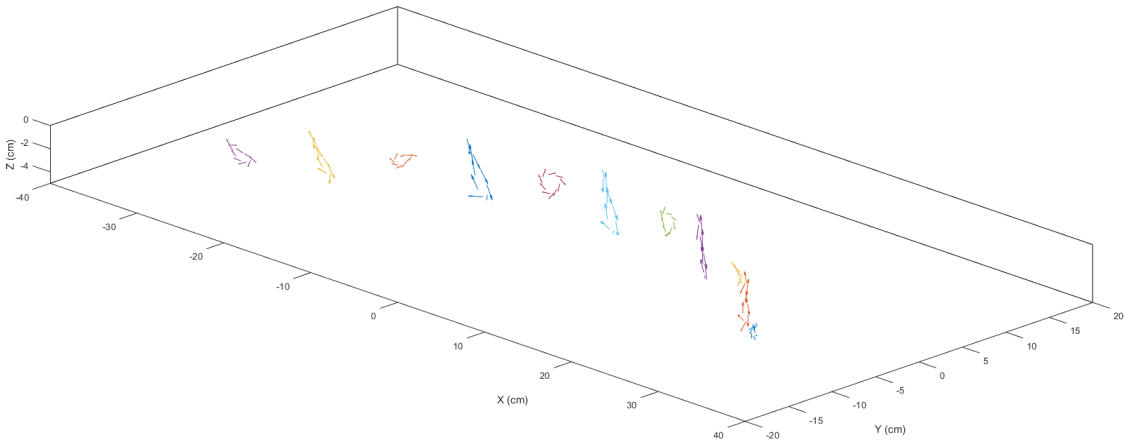
There are several advantages of using the simplified 3D kinematic motion model. Unlike other dynamic models and the robot simulator such as the V-REP, this is less computationally expensive and requires no additional feedback sensors to predict motion trajectories of various gaits except for 2D locomotion gaits such as linear progression and turning gaits [34].

6.4 Kinematic Motion Modelling for 2D Locomotion

As previously discussed, the 3D kinematic motion model proposed by Enner et al. [34] could not predict the motion trajectories for a linear progression gait and a turning gait. Nevertheless, the motion trajectories for these gaits can still be predicted with a slight modification to the model. The external motion can be estimated by assuming that the bottom of the robot's joints rotate in the y-axis similarly to wheels. Unlike the 3D kinematic motion model, a rotation in the internal motions occurs at each pitch joint as shown in Figure 6.7.



(a) Linear progression



(b) Turning

Figure 6.7: The motion traces of the internal motions in a single cycle for linear progression and turning gaits.

To predict the external motions for a linear progression gait and a turning gait, the internal rotational motions about the x and y axes are predictd by using Equation 6.24. ΔX_t and ΔY_t are the motion displacements generated by the robot at time, t, and k is

the number of pitch modules.

$$\begin{bmatrix} \Delta X_t \\ \Delta Y_t \\ 1 \end{bmatrix} = \sum_{k=1}^{n/2+1} \left| \Delta A_t^{2(k-1)+1} \right| \tag{6.24}$$

The external motion of the robot for a linear progression gait is determined by modifying Equation 6.23 as follows:

$$\begin{aligned} \Delta M_t^x &= (\operatorname{sgn}\omega)\Delta X_t \\ \Delta M_t^y &= 0 \\ \Phi_t &= 0 \end{aligned} \tag{6.25}$$

On the other hand, a turning gait requires several steps to predict its motion trajectory. A turning gait with fixed body arc generates a circular motion that depends on the radius of the arc of the robot's body, r_c. The radius of the arc, r_c, can be estimated with two assumptions. The first assumption is that the snake is in a regular polygon shape when O_y is set to a non-zero. The second assumption is that the trajectory of the motion is approximately equal to the radius of a regular polygon.

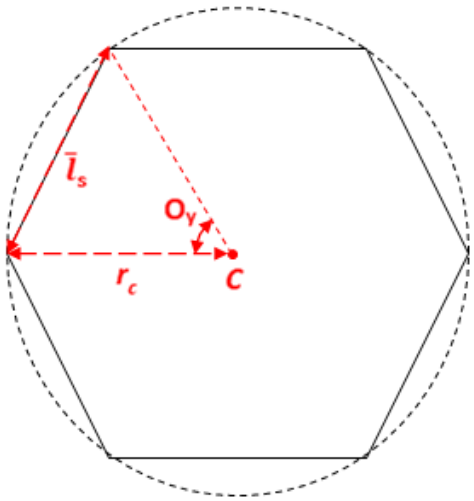


Figure 6.8: The geometrical relationship between a regular polygon and a circle.

As shown in Figure 6.8, r_c is equal to the radius of a regular polygon. The number of vertices of the polygon shape formed by the snake, n_p, is firstly calculated using Equation 6.26. The radius of the arc of the robot's body, r_c, is then estimated using Equation 6.27 [107] where $\bar{l}_s$ is the average of the distances between the joints in the x-axis and the y-axis of the frame.

$$n_c = \frac{360^\circ}{O_y} \tag{6.26}$$

$$r_c \approx \frac{\bar{l}_s}{2sin(\frac{180^\circ}{n_c})} \tag{6.27}$$

The motion displacement of the robot from the point of S_{t-1} and S_t, can be modelled in terms of the arc displacement, ΔS_t as follows:

$$\Delta S_t = \sqrt{\Delta X_t{}^2 + \Delta Y_t{}^2} \tag{6.28}$$

Instead of using the centre of mass of the snake robot in the virtual chassis as the reference point as implemented in the 3D kinematic motion model, the reference point for turning gait is at the fixed point, c, where the snake robot encircles with a fixed radius of r_c as shown in Figure 6.9. The degree of rotation of the robot about the fixed point is determined using Equation 6.29.

$$\Delta\Omega_t = \frac{\Delta S_t}{r_c} \tag{6.29}$$

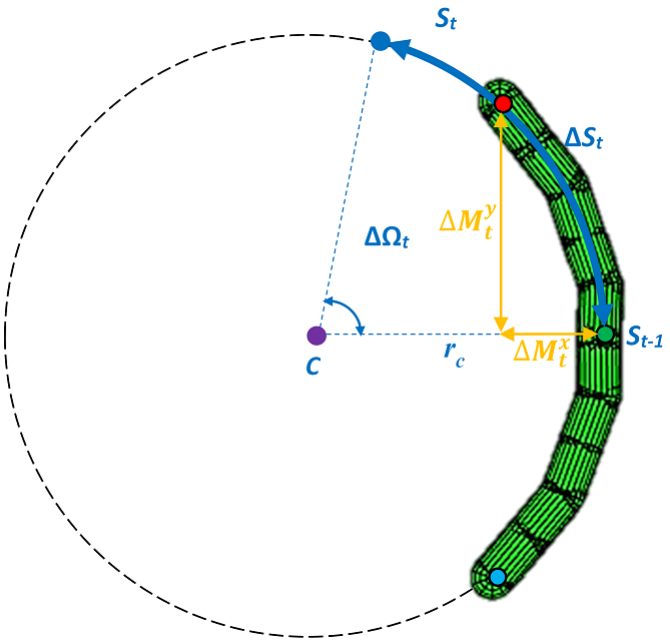


Figure 6.9: The simplified kinematic motion model for turning gait when $O_y > 0$ and $\omega > 0$.

The motion trajectory of the robot in the world frame for a turning gait is estimated by modifying Equation 6.23 as follows:

$$\begin{aligned} \Delta M_t^x &= (\operatorname{sgn}\omega_p) r_c \sin\Delta\Omega_t \\ \Delta M_t^y &= (\operatorname{sgn}\omega_p)\left(r_c - r_c\cos\Delta\Omega_t\right) \\ \Phi_t &= -(\operatorname{sgn}\omega_p)\tan^{-1}\frac{\Delta M_t^y}{\Delta M_t^x} \end{aligned} \tag{6.30}$$

6.5 Experimental Setup

The motion trajectories of the modular snake robot were recorded from the top view of the robot using the set up as in Chapter 5. Figure 6.10 shows one of the frames taken from the recording during the snake's locomotion on a surface board. The recorded frames were then processed using the motion capture developed in the previous chapter to detect the joints of the snake robot. Each trial used different gait parameters, and four trials were conducted for each gait. The set of gait parameters used in each trial for each gait are presented in Tables C.1-C.5 (see Appendix C). Each trial of the experiment contained the motion trajectories of the actual snake robot moving forwards and backwards. The same trials were conducted using the ODE engine in the V-REP simulator. The main purpose of using the simulator was to use its accuracy as a benchmark to compare it against the accuracy of the kinematic motion models. The joint feedback angles recorded from the experiment were used as the input joint angles to generate the motion trajectories in the 2D and the 3D models, and the simulation.

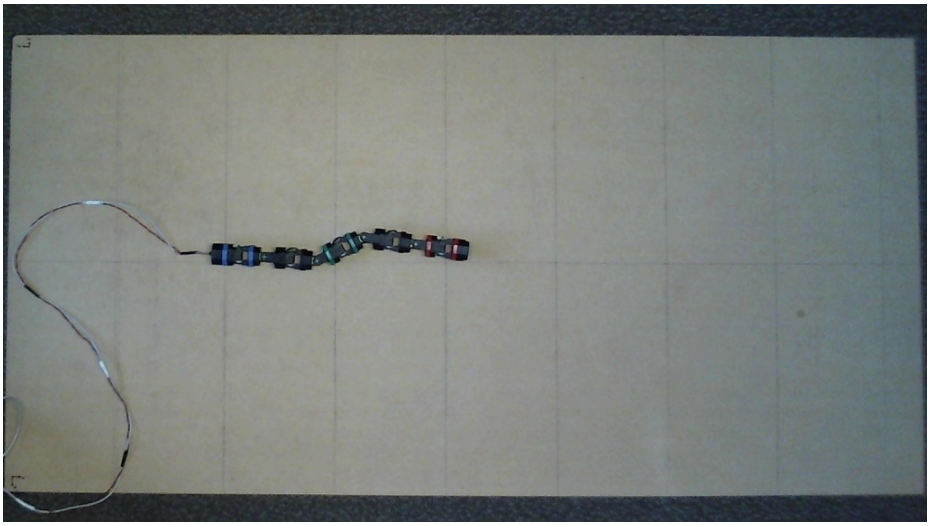
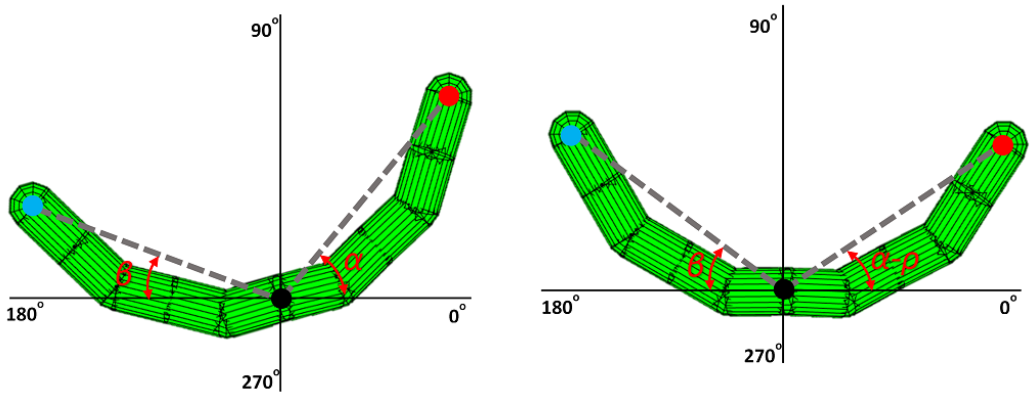


Figure 6.10: A frame taken from the webcam setup to record the motion trajectories of the robot.

6.6 Preprocessing of Motion Trajectory Data

Before the motion trajectories could be optimised and analysed, they were converted from a Cartesian coordinate system to a polar coordinate system. The polar coordinate system describes the errors of the motion trajectories more intuitively in terms of the travelled distance, r, the direction, θ, and the orientation, ϕ, of the snake robot.

The motion trajectories of the experiment and the simulator were then reoriented to match the orientation of the snake robot in the models. This step was crucial to ensure that the errors of the motion trajectories of the robot were calculated accurately. Both the head and the tail modules of the robot were twisted about its centroid until the head and the tail modules were aligned with equal angles from the longitudinal (x) axis, as shown in Figure 6.11b.



(a) Before reorientation ($\beta \neq \alpha$) **(b)** After reorientation ($\beta \approx \alpha - \rho$)

Figure 6.11: Reorientation of the snake robot's body for consistent motion trajectories analysis (the red, black and blue dots are the head, the middle and the tail modules of the robot).

To reorient the robot, the angle required to twist the robot about its centroid was calculated using Equation 6.31. In such a case, as shown in Figure 6.11, the robot was reoriented by twisting the snake in the anti-clockwise direction as $\beta < \alpha$; otherwise, the robot would be twisted in the clockwise direction.

$$\rho = \frac{180^\circ - \beta - \alpha}{2} \tag{6.31}$$

6.7 Consistency of the Motion Trajectories

The consistency of a motion trajectory can be used as an indication of the predictability of the locomotion. The consistency of the motion trajectories of the models were not determined because the random factors, which could influence the motion trajectory of the actual snake robot, cannot be predicted by the models. In addition, the errors in the joint feedback angles would have a minimal impact on the consistency of the predicted motion trajectories because the errors were quite consistent.

6.7.1 Methods

To find the inconsistency in a motion trajectory, the mean motion trajectory was obtained by averaging the motion trajectory in each cycle. The deviation ratio, which is the percentage ratio of the standard deviation and the mean of the motion trajectories, was evaluated to determine the consistency of the motion trajectory. The deviation ratio for each trial of each gait for the experiment and the simulation were determined. The mean of the deviation ratio was determined to find the general consistency of each gait.

6.7.2 Results and Discussion

The mean and the deviation of the motion trajectory for each trial of each gait are presented in Tables D.1-D.5 (see Appendix D). Each trial consists of two sets of data labelled, and the data set for ($\omega > 0$) and ($\omega < 0$) are labelled with 'a' and 'b' respectively. The deviation ratio of each trial for each gait is presented in Tables E.1-E.5 (see Appendix E).

6.7.2.1 Linear Progression Gait

There were two potential factors that affected the consistency of the motion trajectory. One of the factors was the friction generated from the cable attached to the tail module of the robot during the experiment. Since the friction depends on the contact between the cable and the board surface, any inconsistent contact would result in the inconsistency of the motion trajectory. This friction can be assumed to have contributed to the inconsistency of the motion trajectories of other gaits too. Since there was no cable attached to the snake robot in the simulation, the robot in the simulation travelled more consistently than that in the experiment, as shown in Figure 6.12. Another factor was the occurrence of the slips during the locomotion. The slips might have occurred inconsistently during the locomotion due to the hexagonal shape of the modules.

6.7.2.2 Turning Gait

Based on Figure 6.13, r and θ of the experiment and the simulator are less inconsistent than ϕ. The ϕ in both cases are quite consistent as the robot travelled according to the radius of the arc of the robot's body. The consistency of r shows a similar trend to the one for the linear progression gait as both these gaits involve the body extension and contraction during the locomotion. So, the same factors might have affected the

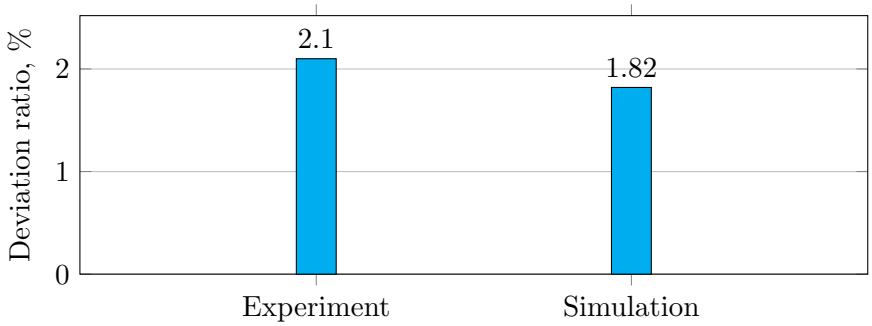


Figure 6.12: The mean of the deviation ratio in the motion trajectories of the linear progression gait trials.

consistency of θ. Figure 6.13 also shows that the outcomes of the simulation are more consistent than those of the experiment.

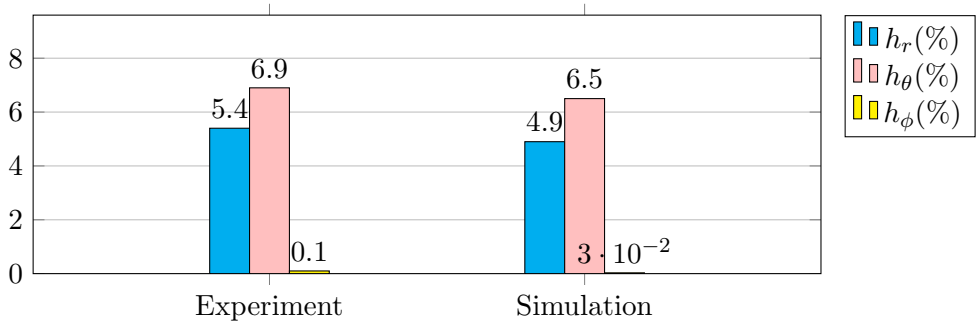


Figure 6.13: The mean of the deviation ratio in the motion trajectories of the turning gait trials.

6.7.2.3 Rolling Gait

Based on Figure 6.14, the outcomes of the experiment and the simulation are more consistent than those of prior gaits because slips are less likely to occur during the rolling gait. However, the inconsistency of ϕ in the experiment is larger than that in simulation due to the presence of a cable at the tail module that might have caused the robot to roll more towards the tail module when the cable was in tension during locomotion [104].

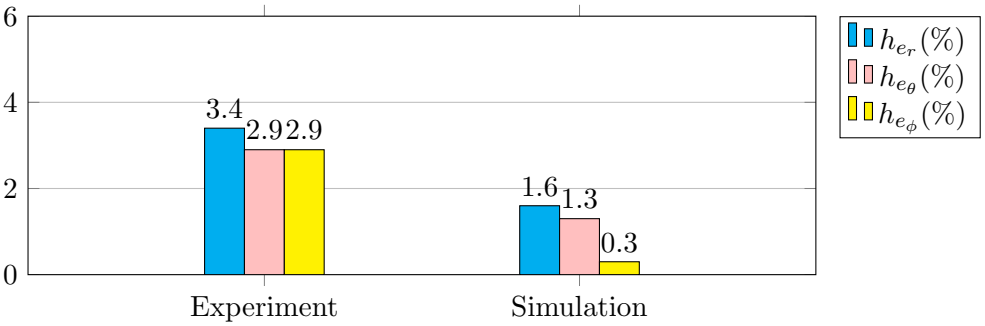


Figure 6.14: The mean of the deviation ratio in the motion trajectories of the rolling gait trials.

6.7.2.4 Rotating Gait

Based on Figure 6.15, the inconsistency of r and θ are significant, indicating that they are not predictable by the 3D model and simulation. In contrast, the ϕ of the experiment and the simulation are very consistent, meaning that the friction produced by the cable was more consistent during the locomotion compared to other gaits.

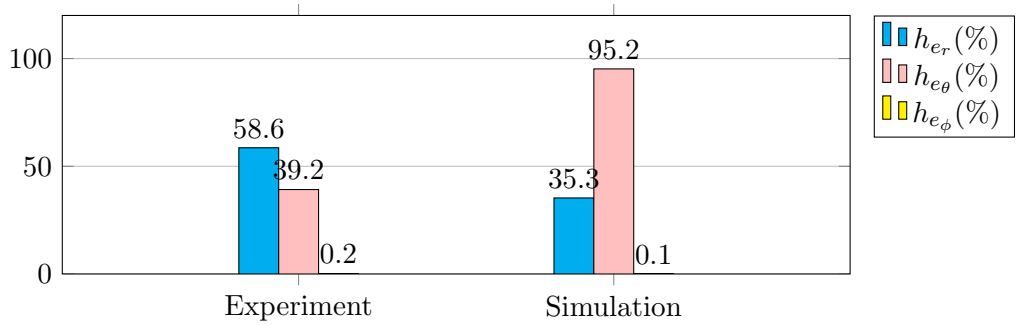


Figure 6.15: The mean of the deviation ratio in the motion trajectories for the rotating gait trials.

6.7.2.5 Sidewinding Gait

Based on Figure 6.16, the inconsistency of r in the experiment and the simulation are significant, meaning that it would be difficult to predict r accurately. Unlike other gaits, the outcomes of the simulation are less consistent than those of the trial. The low consistency in the simulation might be due to the hexagonal shape of the snake robot's modules that caused the ODE engine to compute the interaction between the modules and the ground inaccurately. The hexagonal shape not only affected the outcomes of the simulation but also those of the experiment.

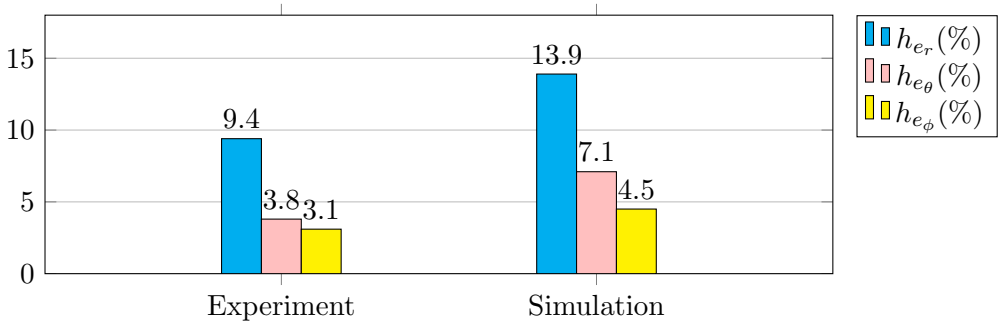


Figure 6.16: The mean of the deviation ratio in the motion trajectories for the sidewinding gait trials.

6.8 Optimisation of 2D and 3D Kinematic Motion Models

The 2D and 3D kinematic motion models need to be optimised to obtain the optimal τ and δ that influence the weight of the contact between the robot's modules and the ground. The optimal τ and δ varied depending on the gait parameters. The average optimal τ and δ can be used to predict the motion trajectory for any gait parameter [34]. In this study, the models were trained using trial data of the experiment to obtain the average optimal τ and δ for each gait. Although using specific optimal τ and δ may give more accurate results, the motion trajectories can still be predicted accurately using the average optimal τ and δ, as presented by Enner et al. [34].

6.8.1 Methods

To optimise τ and δ, the 2D and the 3D kinematics were trained using data of the mean motion trajectories, as presented in Tables D.1-D.5 (see Appendix D). In general, the accuracy of the predicted motion trajectories of the models and the simulation were determined by calculating the errors in terms of the travelled distance, r, direction, θ, and orientation, ϕ, using Equations 6.32-6.34 [34]. However, some of the errors for the turning and the rotating gaits were calculated differently.

$$\varepsilon_r = \frac{r_{pred} - r_{meas}}{r_{meas}} \tag{6.32}$$

$$\varepsilon_\theta = \frac{\theta_{pred} - \theta_{meas}}{\pi} \tag{6.33}$$

$$\varepsilon_\phi = \frac{\phi_{pred} - \phi_{meas}}{\pi} \tag{6.34}$$

Since the robot moves in a circular motion during the turning gait, it is important to ensure that the θ and ϕ of the 3D model follows the circular motion. So, ε_θ and ε_ϕ were calculated using Equations 6.35-6.36

$$\varepsilon_\theta = \frac{\theta_{pred} - \theta_{meas}}{\theta_{meas}} \tag{6.35}$$

$$\varepsilon_\phi = \frac{\phi_{pred} - \phi_{meas}}{\phi_{meas}} \tag{6.36}$$

Since the main purpose of the rotating gait is to rotate the body of the snake robot, the accuracy of the 3D model mainly emphasises e_ϕ. Nevertheless, it was still important to ensure that the middle module stayed within the radius of the robot's initial position so e_r was calculated using Equation 6.37. So, ε_r and ε_ϕ were calculated using Equations 6.37-6.38 where l_d is the length of the module.

$$\varepsilon_r = \frac{r_{pred} - r_{meas}}{l_d} \tag{6.37}$$

$$\varepsilon_{\phi} = \frac{\phi_{pred} - \phi_{meas}}{\phi_{meas}} \tag{6.38}$$

Each set of gait parameters was trained separately using the mean motion trajectory of trial data of the experiment to obtain the total errors, ε_{total}, which were determined using Equation 6.39 [34].

$$\varepsilon_{total} = \sqrt{\lambda_r(\varepsilon_r)^2 + \lambda_\theta(\varepsilon_\theta)^2 + \lambda_\phi(\varepsilon_\phi)^2} \tag{6.39}$$

where λ_r, λ_θ and λ_ϕ are the weight ratios for the travelled distance, travelled direction, and orientation of the snake robot. The weight ratios for each gait were set as presented in Table 6.1. The range of τ and δ used during the training were $[0, 10]$ and $[-15, 15]$. ε_{total} of all the trial data of each gait were averaged, and the lowest point of ε_{total} would be the point where the average optimal τ and δ of the gait is located.

Table 6.1: The weight ratios used for each gait to find optimal τ and δ.

Gait	λ_r	λ_θ	λ_ϕ
Linear progression	1.00	0	0
Turning	0.75	1	1
Rolling	0.20	1	1
Rotating	0.10	0	1
Sidewinding	0.75	1	1

Based on Table 6.1, λ_r for the linear progression gait was set to a high ratio because ε_r is critical for high precision positioning. λ_θ and λ_ϕ were set to zero because θ and ϕ could be predicted by the 2D model. λ_r was not set to zero for the rotating gait to ensure that the robot does not move further away from its original spot. As for the turning gait and the sidewinding gait, λ_r was set to high ratio to ensure that the robot does not just rotate at the same spot but also moves to a new spot. This was because at certain τ and δ, θ and ϕ can be predicted, but the robot may remain at the same spot. As for the rolling gait, λ_r was set at a low ratio because r does not change significantly with the change of τ and δ. In contrast, θ and ϕ are affected drastically with the change of τ and δ. The average optimal τ and δ for each gait was obtained by averaging ε_{total} from all trials.

6.8.2 Results

The average of the optimal τ and δ for each gait (except for sidewinding gait) lies in the dark blue regions in Figures 6.17-6.20. The optimal τ and δ used to predict the motion

trajectories for each gait are presented in Table 6.2. For linear progression and turning gaits, motion trajectories can be computed when $\tau >= 0$.

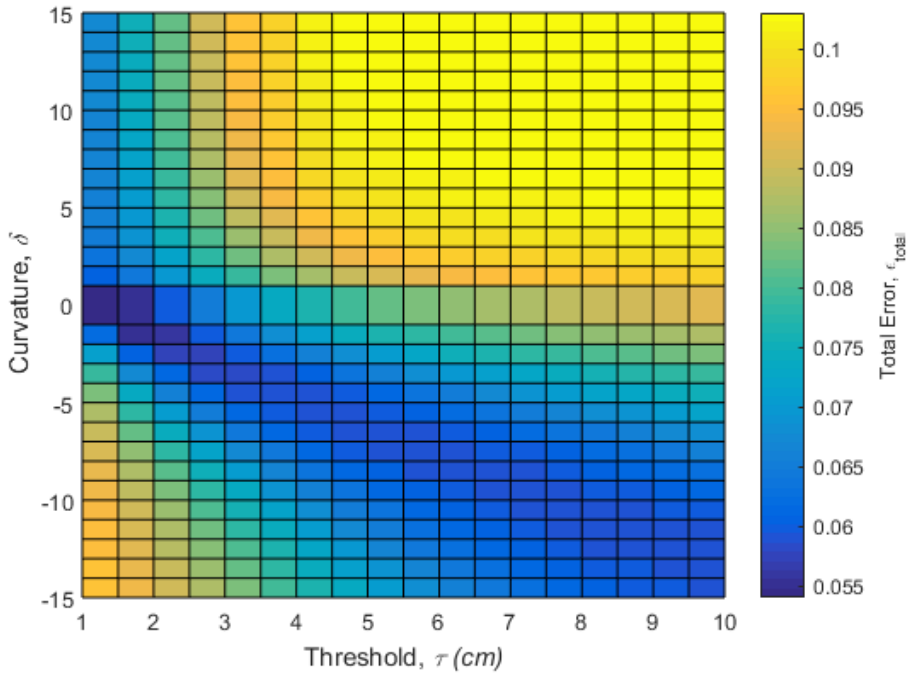


Figure 6.17: The average of the optimal δ and τ for the linear progression trials.

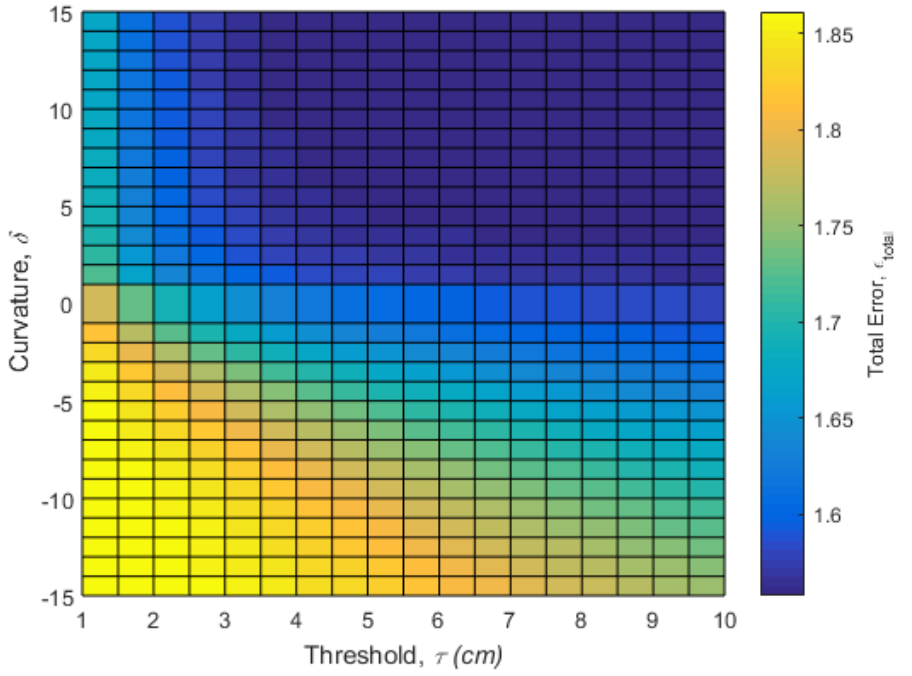


Figure 6.18: The average of the optimal δ and τ for the turning gait trials.

On the other hand, other gaits can only be computed when $\tau >= 3$. Unlike other

gaits, the ground model, which is used in the 2D model, assumes that the ground level of the robot in the virtual chassis starts at the lowest point of the modules.

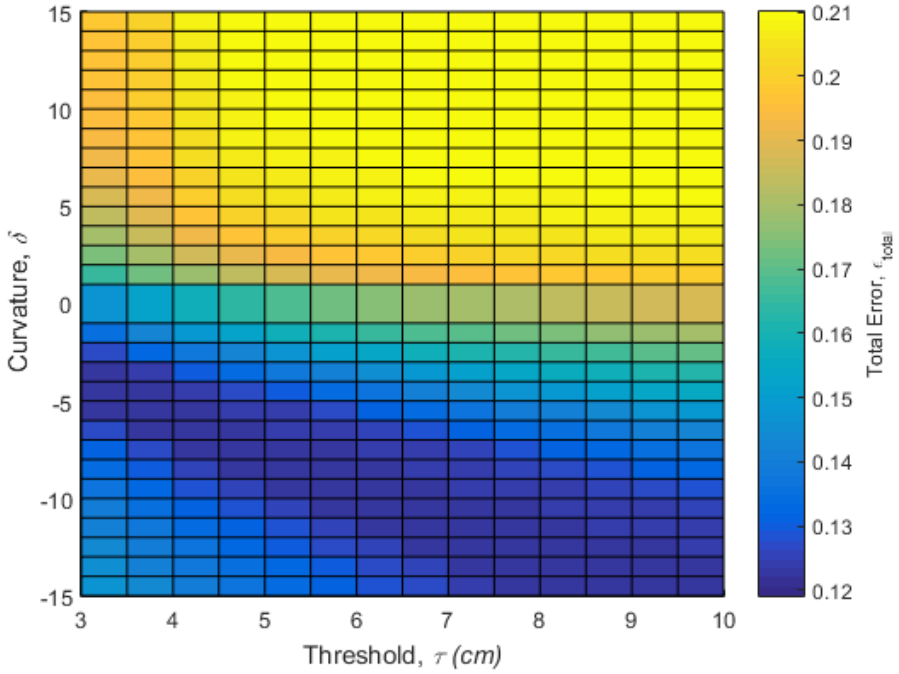


Figure 6.19: The average of the optimal δ and τ for the rolling gait trials.

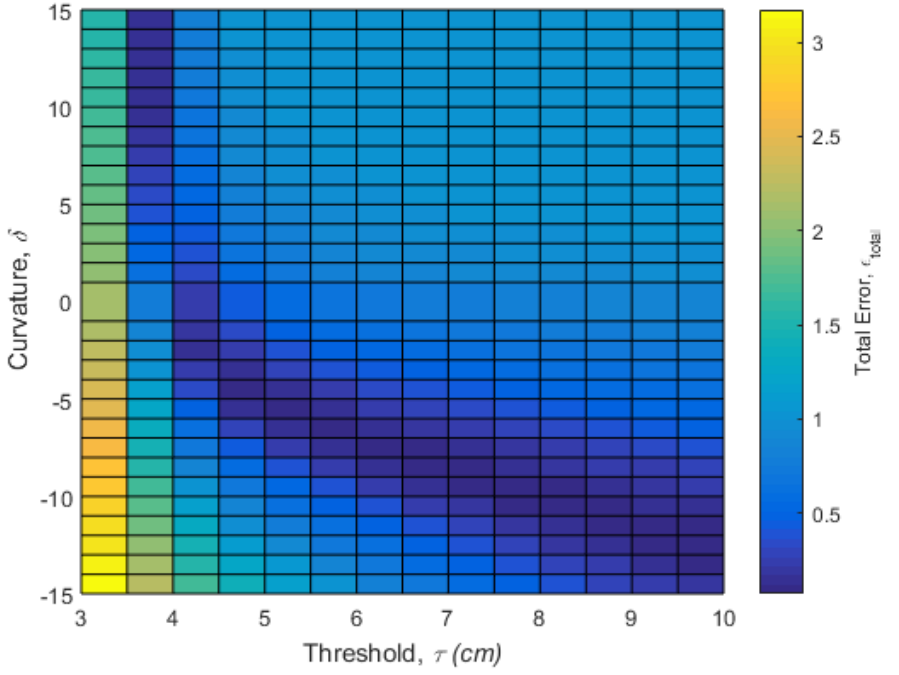


Figure 6.20: The average of the optimal δ and τ for the rotating gait trials.

Table 6.2: The average optimal τ and δ for each gait.

Gait	τ (cm)	δ
Linear progression	1	-1
Turning	5	5
Rolling	3	-5
Rotating	4.5	-5

It was found that the average optimal τ and δ of each trial for the sidewinding gait gave substantial errors as small changes in τ and δ could significantly change the behaviour of the motion trajectory. Thus, the optimal τ and δ for each trial, as presented in Table 6.3, was used to predict the motion trajectory for the sidewinding gait. As for other gaits, the small changes in τ and δ led to some errors in the motion trajectory of each trial, but these errors were not large enough to affect the behaviour of the motion trajectory.

Table 6.3: The optimal τ and δ for each trial of the sidewinding gait.

Trial	τ (cm)	δ
1	3	-15
2	3	15
3	3	-15
4	8.5	-15

6.9 Accuracy of 2D and 3D Kinematic Motion Models

6.9.1 Methods

The motion trajectories of the models were generated using the average optimal δ and τ, as presented in Table 6.2. The raw data of the motion trajectories from the experiment, the models and the simulation are presented in Tables F.1-F.5 (see Appendix F). This data contains the final positions of the robot at 12 seconds while performing the linear progression, turning, rolling and rotating gaits. Since the speed of the sidewinding gait was set to be twice slower than the other gaits, the final positions at 24 seconds are presented. The data was used to determine the errors, ε_r, ε_θ, and ε_ϕ. The errors for each trial of the experiment and the simulation were calculated using Equations 6.32-6.34. The accuracy of the models and the simulation in predicting each gait was evaluated based on the mean of the errors of all the trials of each gait.

6.9.2 Results and Discussion

The errors for each trial of each gait are presented in Tables G.1-G.5 (see Appendix G). In this section, the plot figures of the 'experiment' and 'simulation' are abbreviated as 'Expt' and 'Sim' respectively. 'Head', 'Middle' and 'Tail' are the traces of the head module, the middle module and the tail module of the robot respectively.

The errors in the motion trajectories of the models might be slightly large with the use of the optimal ground parameters, τ and δ. Although the errors in the motion trajectories of the models may be reduced by using specific optimal τ and δ, the reduction of errors would be limited by the inconsistency in the motion trajectories of the gaits of the experiment and the reprojection errors. In general, this factor might have contributed to a certain amount of errors in the motion trajectories of the 2D and the 3D models. However, this might not be the case if the inconsistency of the motion trajectories were caused by the tension in the cable for gaits such as the rolling and the sidewinding, as the tension would be perpendicular to the direction of the external motions of the gaits.

6.9.2.1 Linear Progression Gait

A snake robot ought to move in a straight path while performing linear progression gait. However, this was not the case for the robot in the experiment. The robot in the experiment moved slightly towards the side as shown in Figure 6.21. Figure 6.22 shows that the deviation in the robot's motion in the x-axis gets larger at the end of the cycle. This deviation indicates that the robot did not move perfectly straight as expected from the 2D model and the simulation. The hexagonal shaped module of the robot might have caused each module to grip the ground at inconsistent points of the module, which led to this deviation. Since this deviation is insignificant and cannot be modelled by the 2D model, it is neglected in this study.

Most trials, as presented in Table G.1, present negative errors, meaning that the robot in the experiment moved slightly further than the robot in the simulation and the model. This outcome might have been due to the reprojection errors and from the motion capture. The reprojection errors might have caused the robot to travel roughly 1.6% (the sum of the square root of the reprojection errors in the x-axis and the y-axis) more than it actually did in the experiment. Assuming this was the case, the errors of the simulation in Table G.1 would be more positive, indicating that the robot travelled less further in the experiment than in the simulation. The positive errors might be due to the errors in the feedback

joint angles, friction from the cable, and the slippage that occurred in the experiment.

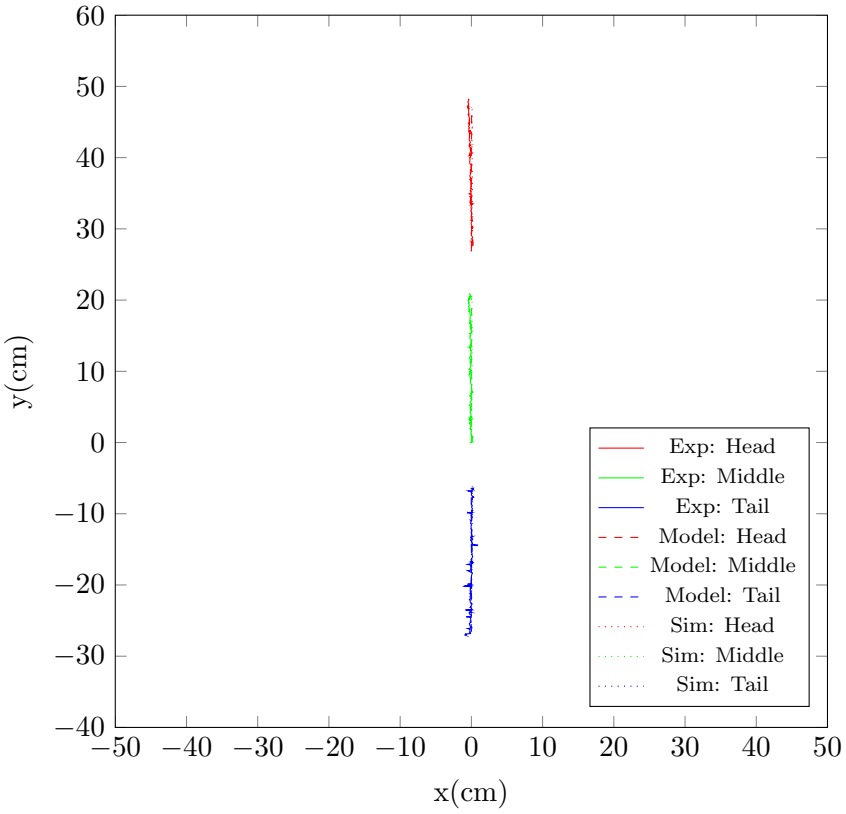


Figure 6.21: The motion trajectories of the linear progression gait, Trial 1a.

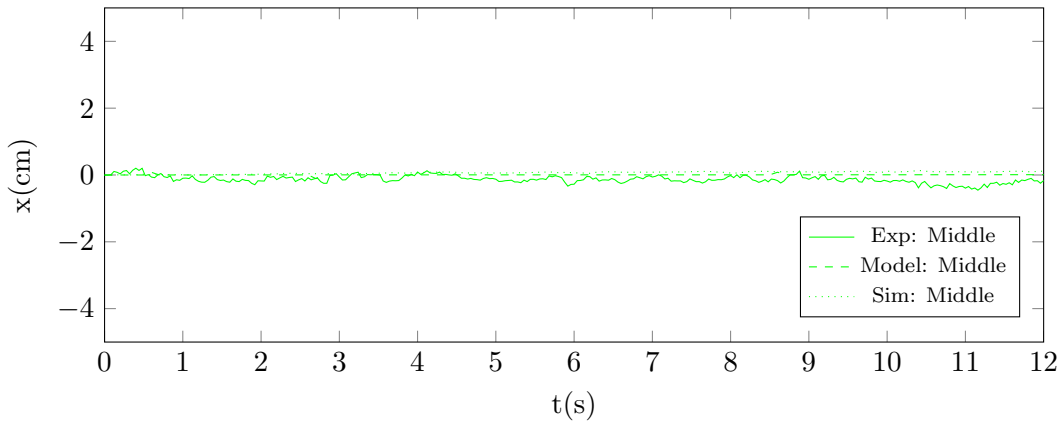


Figure 6.22: The motion trajectory of the linear progression gait, Trial 1a, in the x-axis.

Based on Figure 6.23, the average error in the simulation and the 2D model are about the same. Although the slips are neglected by the 2D model, the inconsistency in the motion trajectories of the gait were not large enough to affect the accuracy of the 2D model when the optimal τ and δ were used.

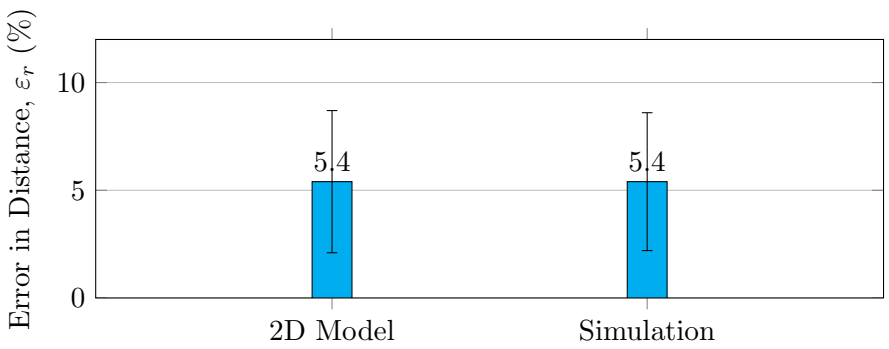


Figure 6.23: The average of the errors in the motion trajectories of the linear progression gait trials.

6.9.2.2 Turning Gait

Since the reprojection error in the x-axis (+1.5%) is larger than the y-axis (+1.2%), the shape of the robot of the experiment appears to have a larger degree of curvature compared to robot of the 2D model and the simulation, as shown in Figure 6.24. This effect may not only contribute to the error in r but in θ and ϕ as well.

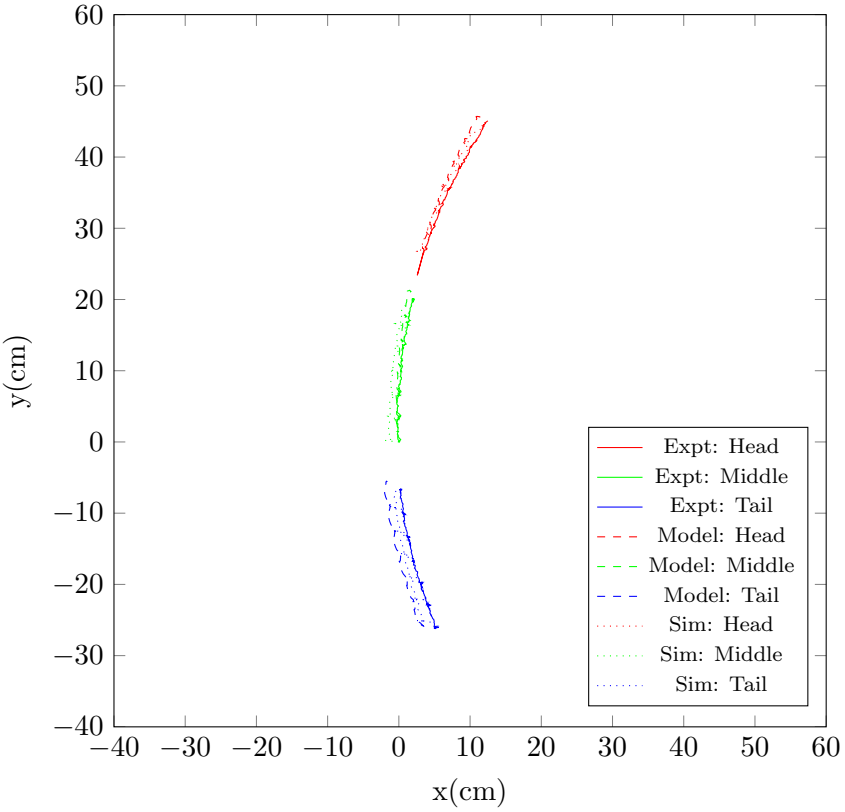


Figure 6.24: The motion trajectories of the turning gait, Trial 1a.

Based on Figure 6.25, the error in r of the simulation shows as a similar trend to that of the linear progression gait. The reprojection errors, friction from the cable, and slippage

might be the factors that contributed to this error. These factors might have contributed to the errors in θ and ϕ as well. On the other hand, the errors of the 2D model are larger than those of the simulation, as shown in Figure 6.25. Based on Table G.2, the errors are significant when $\psi_p < 144°$, indicating the model might not be well optimised when using the average optimal tau and θ for this case.

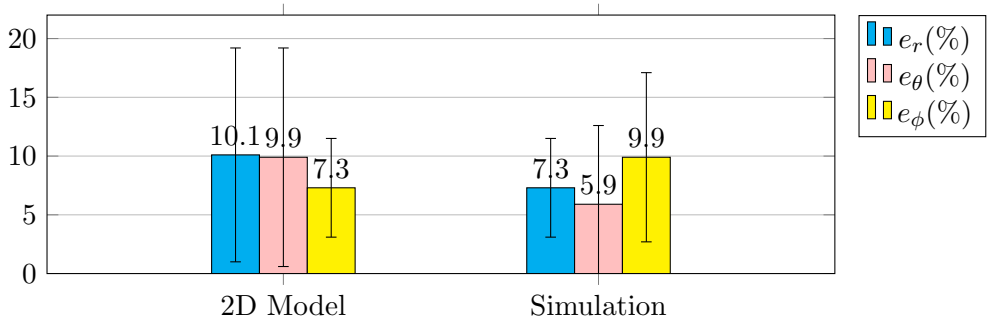


Figure 6.25: The average of the errors in the motion trajectories of the turning gait trials.

6.9.2.3 Rolling Gait

Figure 6.26 shows that the motion trajectories of the rolling gait deviate in the direction towards the centre of mass of the robot. Based on Figure 6.26, the motion trajectory of the robot in the simulation deviates slightly more than that of the robots in the experiment and the 3D model because the simulation modelled the motion differently.

Based on Table G.3, the errors in r of the simulation are in negative values, indicating that the robot in the simulation travelled a shorter distance than the robot in the experiment. These negative errors might be due to the assumption made during the computation performed by the ODE engine. Since pure shapes are highly suggested for a faster and more accurate simulation [108], the computation of the ODE engine might have been affected by the hexagonal shape modules that were imported as compound convex shapes, as shown in Figure A.6 (see Appendix A). The ODE engine might have assumed that the robot was rolling in a similar way to a wheel, and the motion trajectory was computed based on a model of a rolling wheel. Since the circumference of a hexagon is approximately 19.7% greater than a circle, a rolling hexagon travels a distance of 19.7% greater than that of a circle. The 3D model was slightly modified for the rolling gait to consider the rolling hexagon by increasing the generated motion by 19.7%.

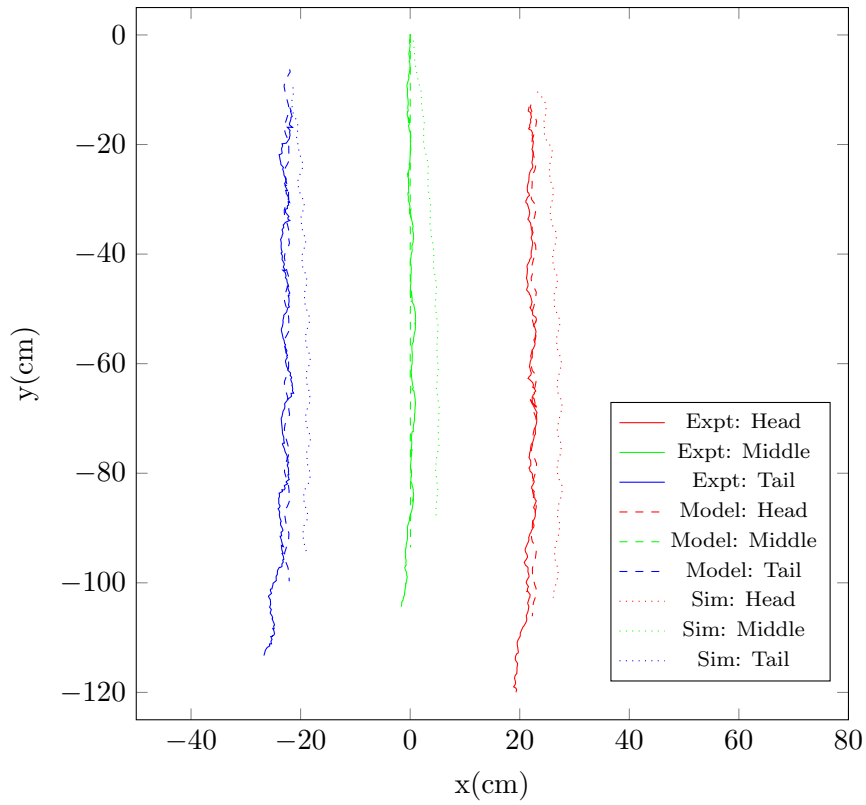


Figure 6.26: The motion trajectories of the rolling gait, Trial 3a.

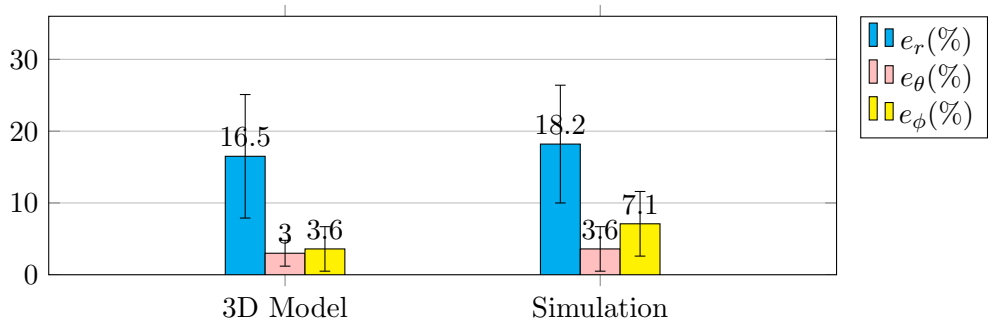


Figure 6.27: The average of the errors in the motion trajectories of the rolling gait trials.

The cable might have caused the motion trajectories to divert more towards the tail module [104]. This diversion, which might have been caused by the tension in the cable during locomotion, resulted in large errors in θ and ϕ in the model and the simulation, as shown in Figure 6.27. The tension in the cable occurred when the robot dragged the cable while rolling. This tension might have led to the errors in θ and ϕ. Based on Table G.3 (see Appendix G), the errors become larger as $A > 30°$ because the 3D model set with the average optimal τ and δ became less accurate in predicting the motion trajectories.

6.9.2.4 Rotating Gait

The motion trajectories of the middle module of the simulation do not overlap with that of the experiment, as shown in Figure 6.28. Based on Figure 6.29, the small errors in r of the 3D model and the simulation indicate that the robot did not move further away from its original spot. Although ϕ of the rotating gait were found to be very consistent in the simulation, the deviation of the errors in ϕ of the simulation is significantly large, as presented in Table G.4. This large deviation might be due to the different levels of slips which occurred with different gait parameters. Slips need to occur to allow the robot to rotate when twisting moment of force is generated. This twisting moment of force is generated when the two halves of the robot's body are pushing in the opposite direction to each other.

Although the 3D model does not consider slips, τ and δ influence the amount of rotation motion generated by the displacement of the modules. The use of the average optimal τ and δ enabled the 3D model to indirectly consider the slips during the locomotion but the small change in the friction might have led to the error in ϕ with different gait parameters. The error in ϕ of the simulation is significant because the friction ratio between the robot's module and the ground in the simulation was different from that in the experiment.

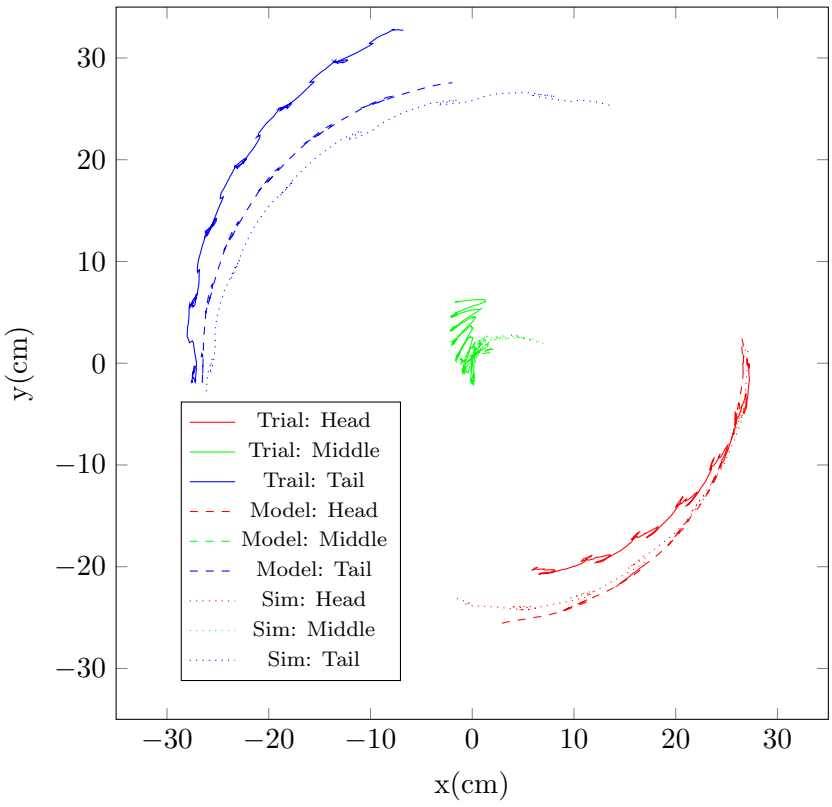


Figure 6.28: The motion trajectories of the rotating gait, Trial 1a.

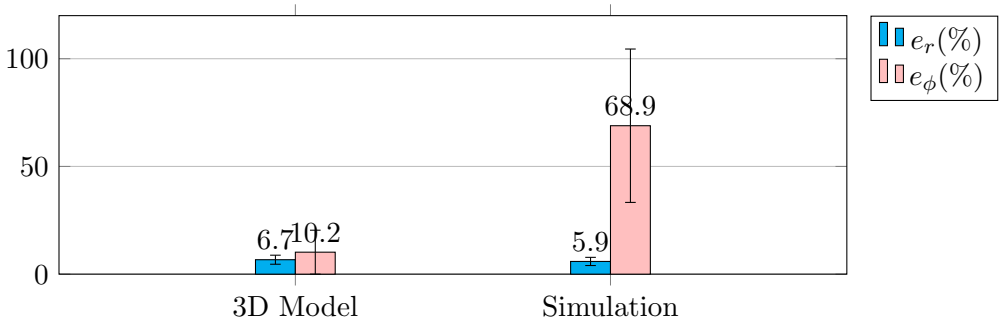


Figure 6.29: The average of the errors in the motion trajectories for the rotating gait trials.

6.9.2.5 Sidewinding Gait

The motion trajectories of the 3D model and the simulation, as shown in Figure 6.30, are overlapping one another, but they divert away from the motion trajectory of the trial. The divergence in the motion trajectory of the simulation indicates that the tension in the cable might have influenced the motion trajectories. The tension in the cable might have again contributed to the large errors in θ and ϕ of the 3D model and the simulation, as shown in Figure 6.31. There are two potential factors that resulted in the large errors in r of the 3D model and the simulation.

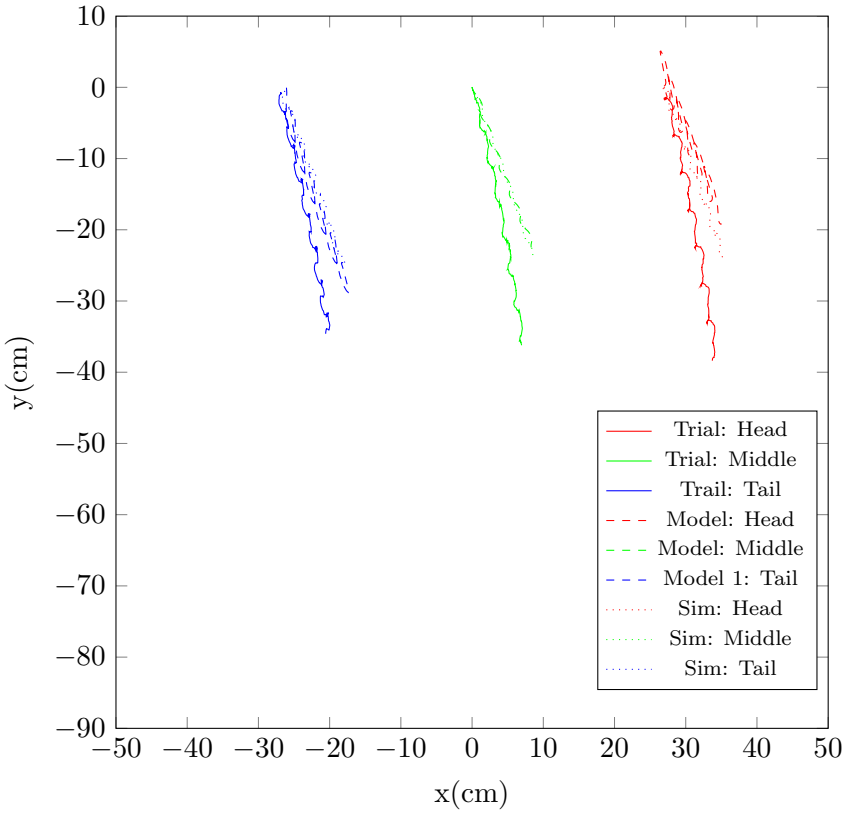


Figure 6.30: The motion trajectories of the sidewinding gait, Trial 1a.

One of the factors is that r of both the 3D model and the simulation were very inconsistent potentially due to the hexagonal shape of the snake robot. The hexagonal shape might have caused the module to have inconsistent contact points with the ground. This shape might have also caused the robot to be slightly unbalanced during the locomotion even though A_p was set to a low value, as was observed during the experiment. This unbalanced locomotion might have caused the robot to move further than predicted by the 3D model and the simulation.

Another factor is the lack of the number of modules of the robot that might have caused the robot to generate 'curvy' sidewinding gait without generating a conical sidewinding gait as proposed by Gong et al. [18]. GCTs of the robot during the gait should be in parallel but the lack of the number of modules might have caused the GCTs to be non-parallel. The 3D model and the simulation might not be able to predict the motion trajectories of this 'curvy' sidewinding gait. This might explain the reason why Enner et al. [34] managed to predict the motion trajectories for the sidewinding gait with the original 3D model because the CMU snake robot consists of 16 modules rather than 10 modules.

Although the errors in the feedback joint angles were significant for the gait, it is difficult to assume that it might have contributed to the errors. Since the joints lifted the robot's body lower than it was supposed to be, the robot should have travelled a smaller distance in the experiment than in the model and in the simulation. However, this was not the case as the outcomes were the opposite.

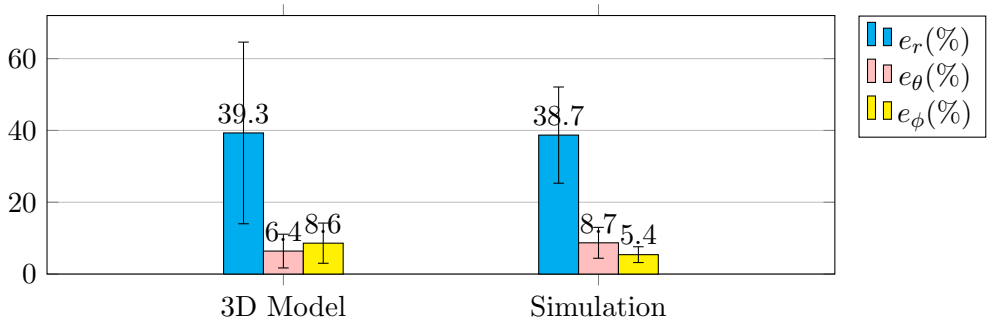


Figure 6.31: The average of the errors in the motion trajectories for the sidewinding gait trials.

6.10 Summary

Four trials with different sets of gait parameters were recorded for each gait. The feedback joint angles were recorded as well and used to generate the motion trajectories of the 2D and the 3D models, and the simulation. The motion trajectories from the experiment and the simulation were aligned to match the initial orientation of the robot in the models.

The consistency of the motion trajectory of each trial for each gait was evaluated to determine its predictability. The linear progression gait and the turning gait are both predictable but might have a little inconsistency due to the cable and the slips which occurred during the locomotion. The inconsistency in the motion trajectories for the sidewinding gait indicated that it would be hard to predict the motion trajectories. This might have been due to the hexagonal shape of the modules. The inconsistency in r and θ are very significant in the rotating gait, indicating that they are not predictable. Nevertheless, ϕ is predictable as it was very consistent.

The 2D and the 3D models were then trained to determine the average optimal τ and δ for each gait. The motion trajectories for most gaits did not change with different gait parameters, but this was not the case for the sidewinding gait. So, instead of using the average optimal τ and δ, specific τ and δ were used for each trial for the gait.

The accuracy of the 2D and 3D models were determined by evaluating the average errors in the predicted motion trajectories for each gait. The errors are presented in terms of r, θ and ϕ. The errors of the simulation were used for comparison. The average optimal τ and δ might have contributed to errors in the motion trajectories of the models. Using specific τ and δ for each trial may improve the accuracy of the models, but this improvement might be limited by the reprojection errors from the motion capture and the inconsistency of the motion trajectories from the experiment. The 2D model predicted the motion trajectories with an error of 5.4%, which is approximately equal to that of the simulation. As for the turning gait, the 2D model predicted the motion trajectories slightly less accurately than the simulation with a highest average error of 10.1%. The highest average error of the 3D model in predicting the motion trajectories for the rolling and the rotating gaits are 16.5% and 10.2% respectively. The 3D model predicted the motion trajectories slightly more accurately than the simulation for the rolling gait. Its predictions for the rotating gait were significantly more accurate than the simulation. Due

to the use of the average optimal τ and δ, the accuracy of the models for the turning gait and the rolling gait dropped drastically with certain gait parameters in the trials. The simulation did not predict the rolling gait accurately as it assumed that the hexagonal module rolls similar to a wheel. The poor predictions of the simulation for the rotating gait might be due to the difference in the friction ratio of the simulation and the experiment. The hexagonal shape of the robot's module and the lack of the number of modules might also have caused the sidewinding gait to be unpredictable in both the model and the simulation.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The design of a modular snake robot took four design iterations for the module to be more versatile, lighter, and easier to assemble. The design was simplified with the use of smart servos. The TTL communication of the robot was reliable up to the frequency of 10 Hz, and the smooth locomotion was generated at the frequency. The locomotion of the robot can be intuitively controlled with the GUI which was based on Qt framework in ROS, but could not switch gaits smoothly. This was not an issue in the thesis as the locomotion during a gait transition was not analysed.

The developed snake robot was capable of performing linear progression, turning, rolling, sidewinding and rotating gaits. The optimal PD gains for the joint control of the robot were obtained and implemented to minimise the errors in the joint angles. Although the joint control was optimised, the errors in the feedback joint angles were reasonably large when the load from the robot's weight exerted on the joints exceeded the power of the servo. The sidewinding gait was greatly affected with the load because there were fewer joints supporting the robot's body during the locomotion. The insufficiency of the servo torque also limited the speed that can be set in the gaits.

To ensure that the robot generates predictable and useful motion trajectories, the optimal gait parameters for each gait were obtained using the ODE engine and validated with the actual snake robot. At certain gait parameters, the lack of the number of modules of the robot and the hexagonal shape of the modules might have resulted in a 'curvy' sidewinding gait. It was also found that the characteristic of the gaits generated by the ODE engine are similar to those in the real world. This finding indicates that the simulator

is suitable to be used for studies on snake robot locomotion. So, the ODE engine can be used to predict the motion trajectories of a modular robot.

A motion capture was developed to track the joints of the robot and reproject their positions into a simulated 3D world. The colour threshold implemented to track the joints sometimes failed due to lighting conditions. Adjusting the colour threshold set in the motion capture resolved the issue. It was found that the reprojection errors in the x-axis and y-axis, which caused the distance measured by the motion capture to be larger than the actual distance, were +1.5% and +1.2%.

The consistency of the motion trajectories, which determine the predictability of the gaits, were evaluated using the mean and the deviation of the motion trajectories. Most motion trajectories of the gaits from the experiment and the simulation were slightly inconsistent mainly due to the tension in the cable and the shape of the modules. However, the inconsistency in the motion trajectories of the sidewinding gait were very significant. The average optimal ground parameters for each gait were obtained by training the models. Since a small change in the ground parameters can lead to substantial differences in the motion trajectories generated for the sidewinding gait, specific ground parameters were used for each trial for the gait. The accuracy was determined by the average of the errors, which are presented in terms of the travelled distance, the travelled direction and the orientation of the robot. The highest average errors of the 2D model for the linear progression and the turning gaits are 5.4% and 10.1% respectively. The prediction of the 2D model was approximate to the simulation for the linear progression gait but slightly less accurate for the turning gait. The highest average errors of the 3D model for the rolling and the rotating gaits are 16.5% and 10.2% respectively. The 3D model managed to predict more accurately than the simulation for the rolling and the rotating gaits. Considering that the two models do not consider slips and require less computation as opposed to the V-REP simulator, they performed reasonably accurately even though their errors are slightly larger than that of the simulation in some trials. In general, the use of the average optimal ground parameters might have contributed to the errors of the models. Although the errors may decrease with the use of specific ground parameters, the decrease might be limited by the reprojection errors and the inconsistency in the motion trajectories. These two factors might have generally affected the accuracy of the models and the simulation for all the gaits. The hexagonal shape of the modules and the lack of the number of modules of the robot might have led the 3D model and the simulation to fail to predict the motion trajectories of the sidewinding gait.

7.2 Future Work

The design of the snake robot needs to be further improved in both the mechanical, and electrical and electronic design aspects. The driving torque of the servo motor used in the existing snake robot was not adequate for the robot to perform pole climbing and faster sidewinding gaits. Increasing the torque more would decrease the limitation of the robot in locomotion. The existing robot could not read the forces exerted on its joints. Elastic joints as proposed by Rollison et al. [65] can be implemented to achieve this capability.

Another necessary improvement is to increase the communication speed of the smart servo motors by replacing the existing TTL communication with RS485 communication. Implementing the tether design suggested by Hirose and Fukushima [68] can minimise the effects of the tether (which is connected to the end of the robot) on the robot's locomotion. Equipping a fully functional camera is challenging because the head module has a limited space. Using a wireless communication for image transmission may help to reduce the load of the RS485 communication, but it is important to be aware of its limitation in the areas where wireless communication is not accessible. Designing intuitive methods of controlling the robot with a remote controller would be a great topic for future studies.

There are a number of improvements required to control the locomotion of the modular snake robot. The existing robot could not switch to a new gait without resetting all of its joint angles to zero. This method is not efficient and practical for field deployment because the robot may not be able to straighten its body in some scenarios. The solution proposed by Droge and Egerstedt [109], which uses a limit cycle method, can be implemented for smooth gait transitions. The existing gait control only allows the robot to perform a 'smooth arc' turning gait which is not suitable for the robot to turn in a sharp corner. To perform a 'sharp' turning gait, the modules of the robot need to bend gradually according to the arch shape that can fit to the corner as the robot travels at the corner [23]. Once the servo motors have a larger torque, the shape of the robot for the pole climbing gait needs to be optimised to ensure the robot grips a pole effectively by using the annealed chain fitting method [49].

The existing motion capture method of the robot can be further developed to operate in real time. However, the algorithm of the motion capture needs to be improved beforehand to minimise the errors in the colour detection method. Replacing the webcam with multiple cameras at different angles would also allow the motion of the robot's joints in the z-axis

to be captured and utilised for future studies on the robot's positioning.

The simplified 2D and the 3D kinematic motion models can be implemented in path-planning, but they should not be used in narrow paths. The errors can be used to estimate how far the robot diverges from the directed path. Further studies need to be conducted to test and validate this implementation. To decrease the divergence of the robot's path from the directed path, the accuracy of the 2D and the 3D models can be further improved. Instead of using the average optimal ground parameters of each gait, specific ground parameters should be used for specific gait parameters. Changing the shape of the modules of the robot to cylindrical shapes and increasing the number of modules of the robot may also help to reduce the inconsistency in the motion trajectories generated by the robot. More studies on their performances with various ground surfaces are required to make them more adaptable to different types of terrain.

References

[1] G. Stylianou, "EQC hires engineering students to help re-inspect Christchurch homes," Christchurch, August 2016. [Online]. Available: http://www.stuff.co.nz/business/82695500/eqc-hires-engineering-students-help-reinspect-christchurch-homes

[2] J. K. Hopkins, B. W. Spranklin, and S. K. Gupta, "A survey of snake-inspired robot designs," *Bioinspiration & biomimetics*, vol. 4, no. 2, p. 21001, 2009.

[3] "New Zealand Earthquakes." [Online]. Available: https://www.gns.cri.nz/Home/Learning/Science-Topics/Earthquakes/New-Zealand-Earthquakes

[4] "Big Pictures: Earthquake again hits Christchurch," February 2011. [Online]. Available: http://www.brisbanetimes.com.au/christchurch-earthquake/

[5] "Photos: Widespread damage after large earthquake hits NZ," November 2016. [Online]. Available: http://www.stuff.co.nz/national/nz-earthquake/86459712/Photos-Widespread-damage-after-large-earthquake-hits-NZ

[6] K. Williams, "How is Wellington coping 100 days on from the 7.8 magnitude Kaikoura quake?" Wellington, February 2017. [Online]. Available: http://www.stuff.co.nz/national/nz-earthquake/89604340/how-is-wellington-coping-100-days-on-from-the-78-magnitude-kaikoura-quake

[7] R. Carmona, D. M. Alba, M. Delfino, J. M. Robles, C. Rotgers, J. V. B. Mengual, J. Balaguer, J. Galindo, and S. Moyà-Solà, "Snake fossil remains from the Middle Miocene stratigraphic series of Abocador de Can Mata (els Hostalets de Pierola, Catalonia, Spain)," *Cidaris*, vol. 30, pp. 77–83, 2010.

[8] A. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, "AmphiBot I: an amphibious snake-like robot," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163–175, 2005.

[9] D. L. Hu and M. Shelley, "Slithering locomotion," in *Natural locomotion in fluids and on surfaces*. Springer, 2012, pp. 117–135.

[10] "Snake Locomotion." [Online]. Available: http://kids.britannica.com/students/assembly/view/171904

[11] D. L. Hu, J. Nirody, T. Scott, M. J. Shelley, and C. S. Peskin, "The mechanics of slithering locomotion." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 25, pp. 10 081–10 085, 2009.

[12] K. Lipkin, I. Brown, A. Peck, H. Choset, J. Rembisz, P. Gianfortoni, and A. Naaktgeboren, "Differentiable and piecewise differentiable gaits for snake robots," in *Proceeding od the 2007 IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 1864–1869.

[13] A. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, "Swimming and crawling with an amphibious snake robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 3024–3028.

[14] M. Mori and S. Hirose, "Three-dimensional serpentine motion and lateral rolling by active cord mechanism ACM-R3," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA)*, vol. 1, 2002, pp. 829–834 vol.1.

[15] S. Yu, S. Ma, B. Li, and Y. Wang, "An amphibious snake-like robot: Design and motion experiments on ground and in water," in *Proceeedings of the 2009 International Conference on Information and Automation (ICIA)*, 2009, pp. 500–505.

[16] A. Shapiro, A. Greenfield, and H. Choset, "Frictional compliance model development and experiments for snake robot climbing," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 574–579.

[17] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset, "Locomotive reduction for snake robots," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3735–3740.

[18] C. Gong, R. L. Hatton, and H. Choset, "Conical sidewinding," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4222–4227.

[19] R. L. Hatton and H. Choset, "Sidewinding on slopes," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 691–696.

[20] H. Marvi, C. Gong, N. Gravish, H. Astley, M. Travers, R. L. Hatton, J. R. Mendelson, H. Choset, D. L. Hu, and D. I. Goldman, "Sidewinding with minimal slip: Snake and robot ascent of sandy slopes," *Science*, vol. 346, no. 6206, pp. 224–229, 2014.

[21] J. Gonzalez-Gomez, H. Zhang, E. Boemo, and J. Zhang, "Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules," in *Proceedings of the 9th International Conference on Climbing and Walking Robots*, 2006.

[22] J. Gonzalez-Gomez, H. Zhang, and E. Boemo, "Locomotion principles of 1D topology pitch and pitch-yaw-connecting modular robots," in *Bioinspiration and Robotics Walking and Climbing Robots*, M. K. Habib, Ed. Vienna: I-Tech Education and Publishing, 2007, pp. 404–428.

[23] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and scripted gaits for modular snake robots," *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, 2009.

[24] K. Melo and L. Paez, "Modular snake robot gaits on horizontal pipes," in *Proceedings of the 2012 IEEE International Conference on Intelligent Robots and Systems (ICRA)*, 2012, pp. 3099–3104.

[25] W. Zhen, C. Gong, and H. Choset, "Modeling rolling gaits of a snake robot," *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2015-June, no. e, pp. 3741–3746, 2015.

[26] S. Hirose, *Biologically inspired robots: snake-like locomotors and manipulators*. Oxford University Press, 1993.

[27] S. Ma, "Analysis of snake movement forms for realization of snake-like robots," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4. IEEE, 1999, pp. 3007–3013 vol.4.

[28] Shugen, "Analysis of creeping locomotion of a snake-like robot," *Advanced Robotics*, vol. 15, no. 2, pp. 205–224, 2001.

[29] G. P. Hicks, "Modeling and control of a snake-like serial-link structure," Ph.D. dissertation, North Carolina State University, 2003.

[30] S. Ma and N. Tadokoro, "Analysis of creeping locomotion of a snake-like robot on a slope," *Autonomous Robots*, vol. 20, no. 1, pp. 15–23, 2006.

[31] S. Ma, Y. Ohmameuda, and K. Inoue, "Dynamic analysis of 3-dimensional snake robots," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2004, pp. 767–772.

[32] P. Liljebäck, Ø. Stavdahl, and K. Y. Pettersen, "Modular pneumatic snake robot: 3D modelling, implementation and control," *Modeling, Identification and Control*, vol. 29, no. 1, pp. 21–28, 2008.

[33] D. Rollinson and H. Choset, "Virtual chassis for snake robots," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 221–226.

[34] F. Enner, D. Rollinson, and H. Choset, "Simplified motion modeling for snake robots," in *Proceedings of 2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 4216–4221.

[35] K. Melo, "Modular snake robot velocity for side-winding gaits," in *Proceeding of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3716–3722.

[36] P. Prautsch and T. Mita, "Control and analysis of the gait of snake robots," in *Proceedings of the 1999 IEEE International Conference on Control Applications (CCA)*, vol. 1, 1999, pp. 502–507 vol. 1.

[37] P. Liljebäck, K. Y. Pettersen, O. Stavdahl, and J. T. Gravdahl, "Controllability and stability analysis of planar snake robot locomotion," *Proceeding of the 2011 IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1365–1380, 2011.

[38] Z. Y. Bayraktaroglu, "Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 591–602, 2009.

[39] T. Kano and A. Ishiguro, "Obstacles are beneficial to me! Scaffold-based locomotion of a snake-like robot using decentralized control," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 3273–3278.

[40] A. A. Transeth, R. Leine, C. Glocker, K. Y. Pettersen, and P. Liljeback, "Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 88–104, 2008.

[41] A. M. Andruska and K. S. Peterson, "Control of a snake-like robot in an elastically deformable channel," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 2, pp. 219–227, 2008.

[42] D. Rollinson, K. V. Alwala, N. Zevallos, and H. Choset, "Torque control strategies for snake robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014, pp. 1093–1099.

[43] D. Rollinson and H. Choset, "Gait-based compliant control for snake robots," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 5138–5143.

[44] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological cybernetics*, vol. 56, no. 5-6, pp. 345–353, 1987.

[45] K. Inoue, M. Shugen, and J. Chenghua, "Neural oscillator network-based controller for meandering locomotion of snake-like robots," in *Proceedings 2004 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 5, 2004, pp. 5064–5069.

[46] X. Wu and S. Ma, "CPG-based control of serpentine locomotion of a snake-like robot," *Mechatronics*, vol. 20, no. 2, pp. 326–334, 2010.

[47] ——, "Head-navigated locomotion of a snake-like robot for its autonomous obstacle avoidance," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 401–406.

[48] K. Inoue, T. Sumi, and M. Shugen, "CPG-based control of a simulated snake-like robot adaptable to changing ground friction," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1957–1962.

[49] R. L. Hatton and H. Choset, "Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction," *Autonomous Robots*, vol. 28, no. 3, pp. 271–281, 2010.

[50] R. L. Hatton, R. A. Knepper, H. Choset, D. Rollinson, C. Gong, and E. Galceran, "Snakes on a plan: Toward combining planning and control," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 5174–5181.

[51] J. K. Hopkins and S. K. Gupta, "Design and modeling of a new drive system and exaggerated rectilinear-gait for a snake-inspired robot," *Journal of Mechanisms and Robotics*, vol. 6, no. 2, p. 21001, 2014.

[52] A. Crespi and A. J. Ijspeert, "AmphiBot II: An amphibious snake robot that crawls and swims using a central pattern generator," in *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, 2006, pp. 19–27.

[53] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, "Design of a modular snake robot," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 2609–2614.

[54] B. Klaassen and K. L. Paap, "Gmd-snake2: a snake-like robot driven by wheels and a method for motion control," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4. IEEE, 1999, pp. 3014–3019.

[55] J. Borenstein and A. Borrell, "The OmniTread OT-4 serpentine robot," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1766–1767.

[56] H. Ohno and S. Hirose, "Study on slime robot (proposal of slime robot and design of slim\nslime robot)," *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2218–2223, 2000.

[57] A. Johnson, C. Wright, M. Tesch, K. Lipkin, and H. Choset, "A novel architecture for modular snake robots," Robotics Institute, Tech. Rep., 2011.

[58] H. Yamada and S. Hirose, "Study of a 2-DOF joint for the small active cord mechanism," in *Proceeding of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3827–3832.

[59] H. Yamada, S. Chigisaki, M. Mori, K. Takita, K. Ogami, and S. Hirose, "Development of amphibious snake-like robot ACM-R5," *Industrial Robot: An International Journal (ISR)*, 2005.

[60] J. C. McKenna, D. J. Anhalt, F. M. Bronson, H. B. Brown, M. Schwerin, E. Shammas, and H. Choset, "Toroidal skin drive for snake robot locomotion," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1150–1155.

[61] A. Wolf, H. B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, and H. Choset, "A mobile hyper redundant mechanism for search and rescue tasks," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3. IEEE, 2003, pp. 2889–2895.

[62] M. Mori and S. Hirose, "Development of active cord mechanism ACM-R3 with agile 3D mobility," in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*, vol. 3. IEEE, 2001, pp. 1552–1557.

[63] H. Yamada and S. Hirose, "Development of practical 3-dimensional active cord mechanism ACM-R4," *Journal of Robotics and Mechatronics*, vol. 18, no. 3, p. 305, 2006.

[64] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, and H. Choset, "Design and architecture of the unified modular snake robot," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 4347–4354.

[65] D. Rollinson, Y. Bilgen, B. Brown, F. Enner, S. Ford, C. Layton, J. Rembisz, M. Schwerin, A. Willig, and P. Velagapudi, "Design and architecture of a series elastic snake robot," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014, pp. 4630–4636.

[66] T. Kamegawa, T. Harada, and A. Gofuku, "Realization of cylinder climbing locomotion with helical form by a snake robot with passive wheels," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3067–3072.

[67] "Hyper redundant snake robot." [Online]. Available: https://brml.technion.ac.il/show{_}project/8{#}prettyPhoto

[68] S. Hirose and E. Fukushima, "Snakes and strings: new robotic components for rescue operations," *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, vol. 1, pp. 2–12, 2002.

[69] P. Liljeback, O. Stavdahl, and A. Beitnes, "SnakeFighter - Development of a Water Hydraulic Fire Fighting Snake Robot," in *Proceedings of the 2006 9th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2006, pp. 1–6.

[70] P. Liljeback, O. Stavdahl, K. Y. Pettersen, and J. T. Gravdahl, "Mamba - A waterproof snake robot with tactile sensing," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 294–301.

[71] M. Mutlu, K. Melo, M. Vespignani, A. Bernardino, and A. J. Ijspeert, "Where to place cameras on a snake robot: Focus on camera trajectory and motion blur," in *Proceedings of the 2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016.

[72] H. Ponte, M. Queenan, C. Gong, C. Mertz, M. Travers, F. Enner, M. Hebert, and H. Choset, "Visual sensing for developing autonomous behavior in snake robots," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2779–2784.

[73] J. Gonzalez-Gomez, J. Gonzalez-Quijano, H. Zhang, and M. Abderrahim, "Toward the sense of touch in snake modular robots for search and rescue operations," in *Proc. ICRA 2010 Workshop Modular Robots: State of the Art*, 2010, pp. 63–68.

[74] P. Liljebäck, S. Fjerdingen, K. Y. Pettersen, and Ø. Stavdahl, "A snake robot joint mechanism with a contact force measurement system," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3815–3820.

[75] D. Rollinson, A. Buchan, and H. Choset, "State estimation for snake robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1075–1080.

[76] M. Sato, M. Fukaya, and T. Iwasaki, "Serpentine locomotion with robotic snakes," *IEEE Control Systems*, vol. 22, no. 1, pp. 64–81, feb 2002.

[77] "HiBot Products: Robots," 2012. [Online]. Available: https://www.hibot.co.jp/en/products/robots{_}1

[78] S. Takaoka, H. Yamada, and S. Hirose, "Snake-like active wheel robot ACM-R4.1 with joint torque sensor and limiter," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1081–1086.

[79] P. Liljebäck, K. Y. Pettersen, O. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control*. Springer Science & Business Media, 2012.

[80] S. A. Fjerdingen, P. Liljebäck, and A. A. Transeth, "A snake-like robot for internal inspection of complex pipe structures (PIKo)," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 5665–5671.

[81] A. Ertas and J. C. Jones, *The engineering design process*. Wiley New York, 1996.

[82] K. Melo, L. Paez, and C. Parra, "Indoor and outdoor parametrized gait execution with modular snake robots," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, 2012.

[83] H. Yamada, S. Takaoka, and S. Hirose, "A snake-like robot for real-world inspection applications (the design and control of a practical active cord mechanism)," *Advanced Robotics*, vol. 27, no. 1, pp. 47–60, 2013.

[84] E. Canessa, C. Fonda, and M. Zennaro, "Low–cost 3D printing for science, education and sustainable development," *Low-Cost 3D Printing*, vol. 11, 2013.

[85] T. Letcher and M. Waytashek, "Material Property Testing of 3D-Printed Specimen in PLA on an Entry-Level 3D Printer," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2014.

[86] S. Farah, D. G. Anderson, and R. Langer, "Physical and mechanical properties of PLA, and their functions in widespread applications—a comprehensive review," *Advanced drug delivery reviews*, vol. 107, pp. 367–392, 2016.

[87] M. Page, "Understanding D.C. Motor Characteristics," 1999. [Online]. Available: http://lancet.mit.edu/motors/motors3.html

[88] "Robot Operating System (ROS)," 2016. [Online]. Available: http://www.ros.org/

[89] "rqt," 2016. [Online]. Available: http://wiki.ros.org/rqt

[90] "Virtual Robot Experimentation Platform," 2017. [Online]. Available: http://www.coppeliarobotics.com/

[91] J. Burdick, J. Radford, and G. Chirikjian, "A 'sidewinding' locomotion gait for hyper-redundant robots," *Proceedings of the 1993 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–6, 1993.

[92] M. Field, D. Stirling, F. Naghdy, and Z. Pan, "Motion capture in robotics review," in *Proceedings of the 2009 IEEE International Conference on Control and Automation (ICCA)*, 2009, pp. 1697–1702.

[93] H.-D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.

[94] D. J. Bora, A. K. Gupta, and F. A. Khan, "Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation," *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, no. 2, pp. 192–203, 2015.

[95] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," *International Conference on Image Processing*, vol. 2, pp. II–589–II–592, 2002. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1040019

[96] D. Khattab, H. M. Ebied, A. S. Hussein, and M. F. Tolba, "Color image segmentation based on different color space models using automatic GrabCut." *The Scientific World Journal*, vol. 2014, p. 126025, 2014. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84921413177{&}partnerID=tZOtx3y1

[97] X. Wang, R. Hänsch, L. Ma, and O. Hellwich, "Comparison of different color spaces for image segmentation using graph-cut," in *Proceedings of the 2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 1. IEEE, 2014, pp. 301–308.

[98] J. Ponce and D. Forsyth, *Computer vision: a modern approach.* Pearson, 2012. [Online]. Available: http://www.inria.fr/centre/paris-rocquencourt/actualites/computer-vision-a-modern-approach

[99] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[100] J.-Y. Bouguet, "Complete Camera Calibration Toolbox for Matlab," 1999. [Online]. Available: http://www.vision.caltech.edu/bouguetj/

[101] "Evaluating the Accuracy of Single Camera Calibration," 2016. [Online]. Available: https://au.mathworks.com/help/vision/examples/evaluating-the-accuracy-of-single-camera-calibration.html

[102] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[103] "Measuring Planar Objects with a Calibrated Camera." [Online]. Available: https://au.mathworks.com/help/vision/examples/measuring-planar-objects-with-a-calibrated-camera.html

[104] C. Gilani, X. Chen, C. Pretty, and C. Koike, "Visualisation of the Motion Trajectory for Rolling Motion of Snake Robots using Virtual Chassis and Simplified Kinematics Motion Model," *IFAC-PapersOnLine*, vol. 49, no. 32, 2016.

[105] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control.* Springer-Verlag London, 2009.

[106] R. Bro, E. Acar, and T. G. Kolda, "Resolving the sign ambiguity in the singular value decomposition," *Journal of Chemometrics*, vol. 22, no. 2, pp. 135–140, 2008.

[107] J. D. Page, "Regular Polygon." [Online]. Available: http://www.mathopenref.com/polygonregular.html

[108] "Shapes," 2017. [Online]. Available: http://www.coppeliarobotics.com/helpFiles/en/shapes.htm

[109] G. Droge and M. Egerstedt, "Optimal decentralized gait transitions for snake robots," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 317–322.

Appendices

A V-REP Simulator Settings

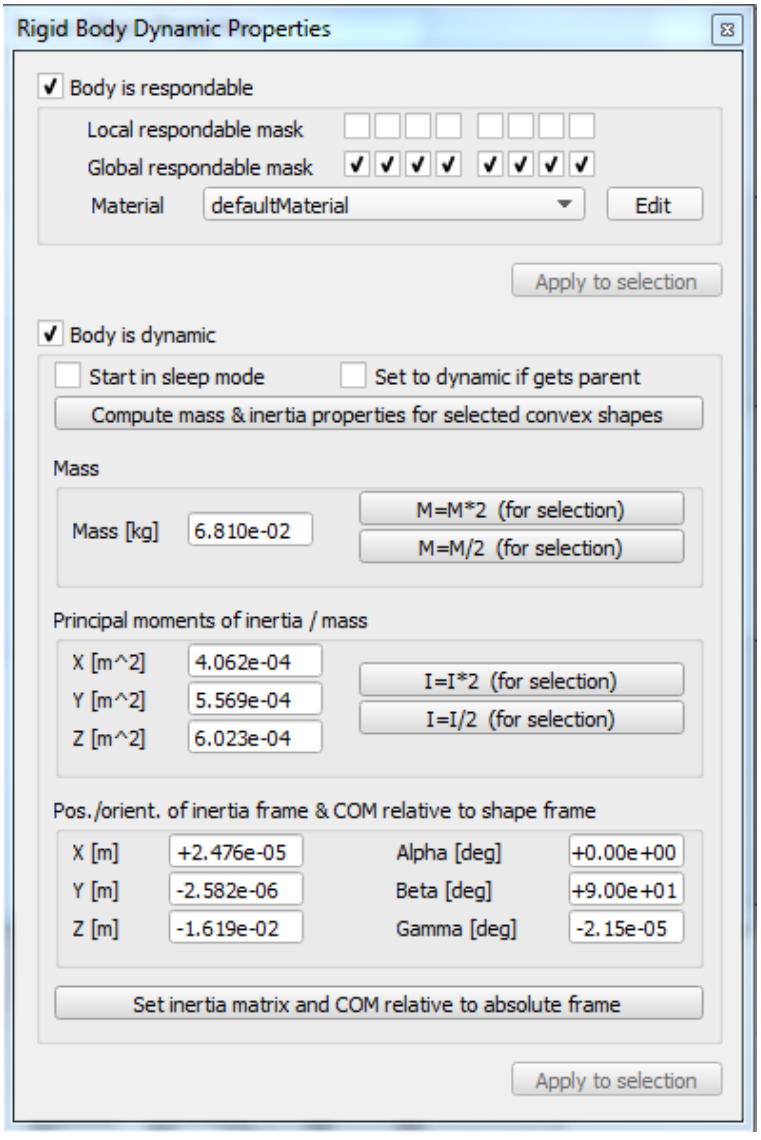


Figure A.1: The settings of the head module of the imported snake robot in the simulator.

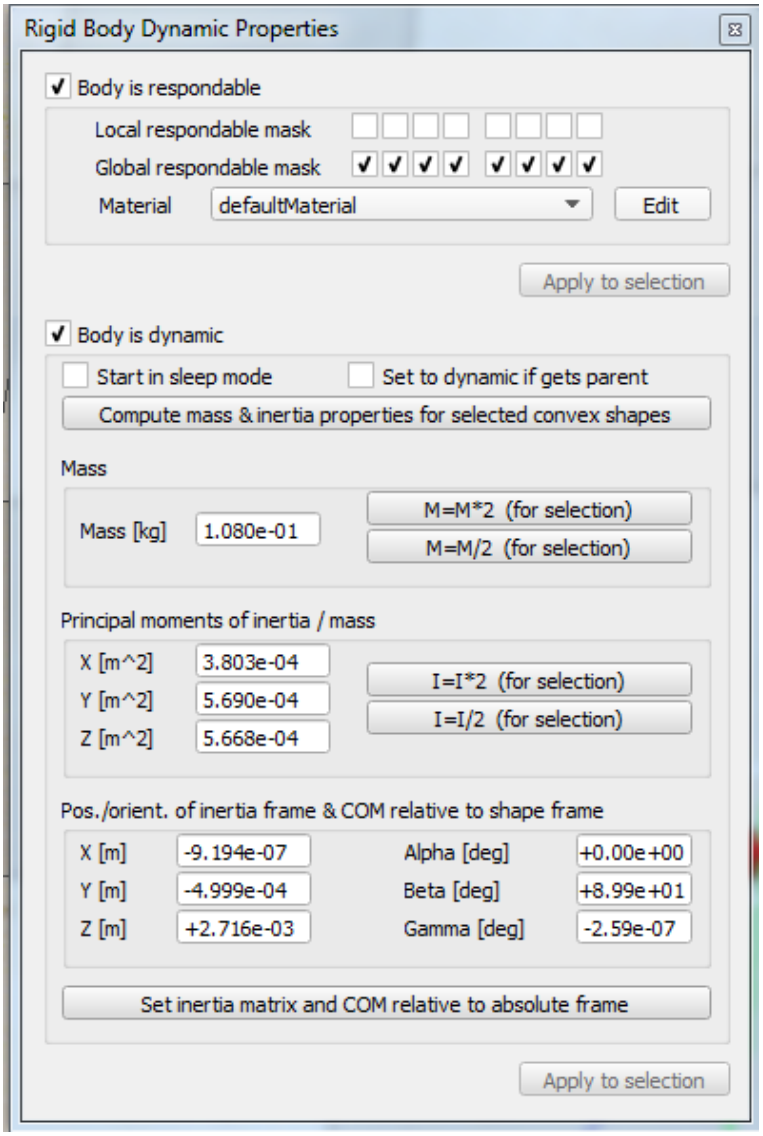


Figure A.2: The settings of the tail module of the imported modular snake robot in the simulator.

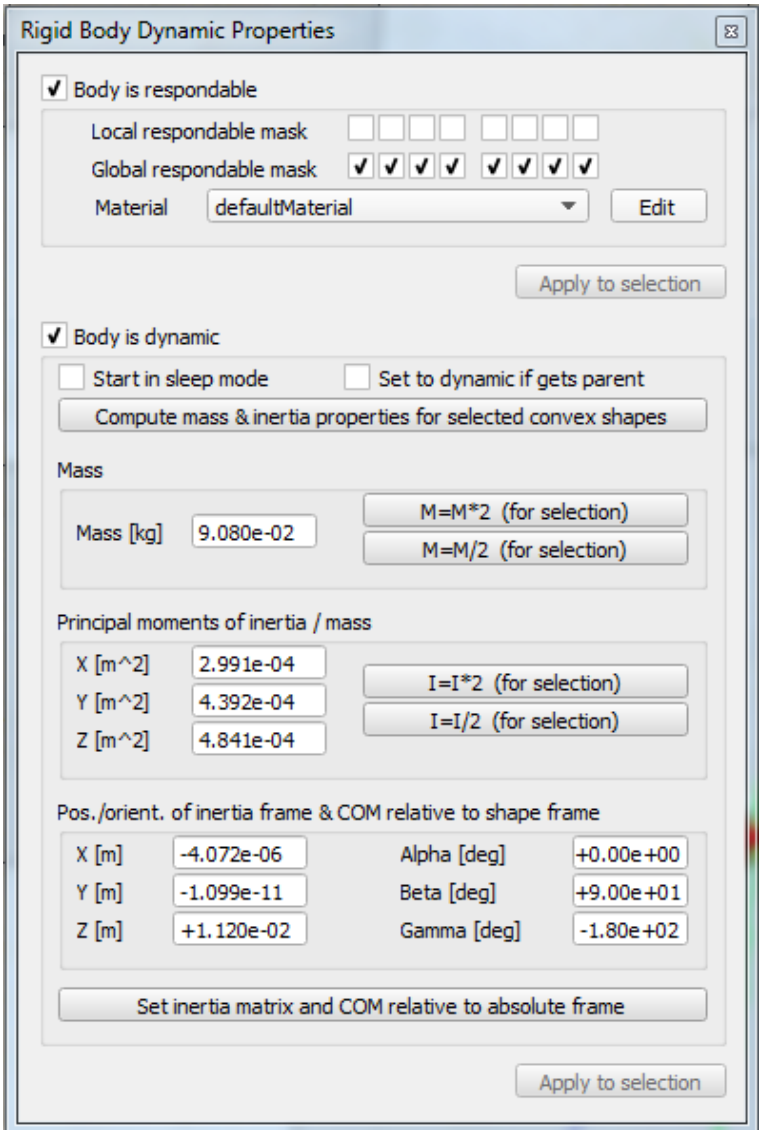


Figure A.3: The settings of each regular module of the imported modular snake robot in the simulator.

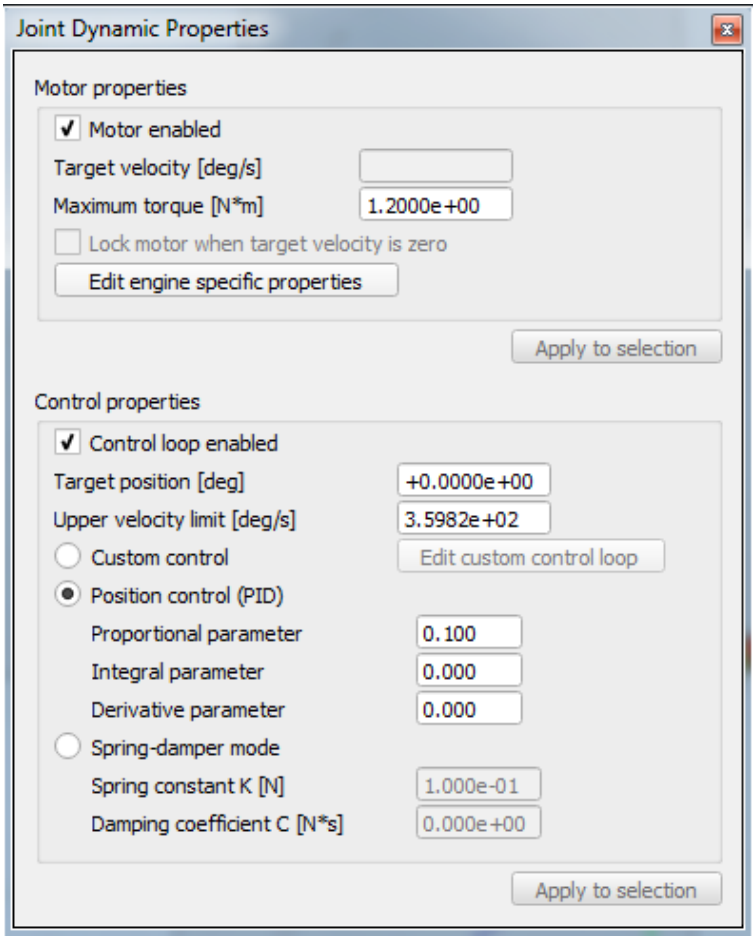


Figure A.4: The settings of each joint of the imported modular snake robot in the simulator.

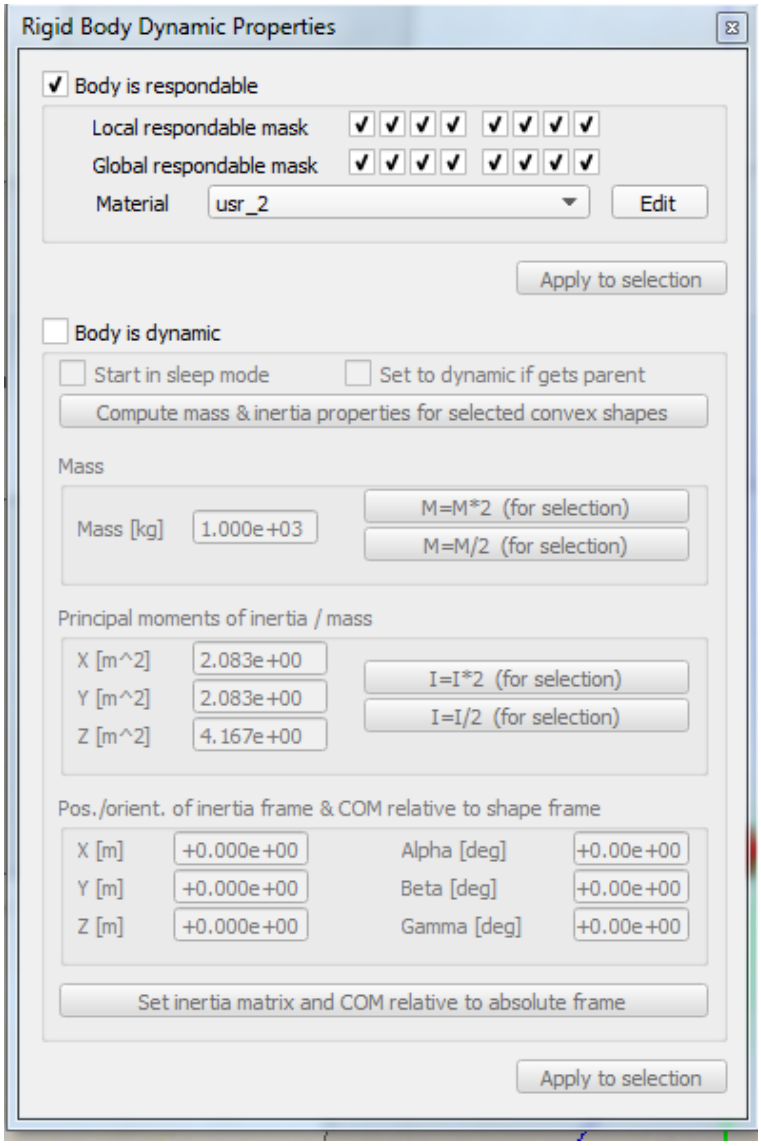


Figure A.5: The settings of the floor in the simulator.

Figure A.6: The modules of the snake robot are imported as compound convex shapes.

B Reprojection Errors of the Motion Capture

Table B.1: The coordinates of each pair of red, green and blue markers in the simulated 3D world coordinates.

Line	Red				Greeen				Blue			
	x1 (mm)	x2 (mm)	y1 (mm)	y2 (mm)	y1 (mm)	x2 (mm)	y1 (mm)	y2 (mm)	X1 (mm)	X2 (mm)	Y1 (mm)	Y2 (mm)
1	295.4	226.0	188.3	188.8	20.4	-52.9	192.0	187.7	-257.6	-330.2	182.9	183.3
2	296.0	226.8	482.7	486.2	16.8	-49.3	480.1	483.7	-258.4	-328.6	476.7	477.9
3	291.3	221.8	777.3	777.4	15.5	-52.8	773.2	770.5	-331.5	-261.5	766.8	768.3
4	292.3	223.1	1071.3	1076.0	14.3	-53.9	1066.4	1066.9	-332.5	-262.3	1061.7	1063.6
5	287.8	222.1	1353.8	1361.8	15.4	-55.1	1356.4	1356.4	-263.1	-331.7	1354.4	1353.3
6	282.6	215.7	1651.2	1652.9	11.8	-58.5	1651.1	1648.5	-335.2	-267.4	1640.2	1648.8
7	281.7	215.2	1922.5	1938.8	8.2	-59.7	1937.8	1941.1	-337.7	-266.8	1929.8	1935.5

Table B.2: The mid points coordinates of each pair of red, green and blue markers in the simulated 3D world coordinates.

Line	Red		Green		Blue	
	Midpoint X (mm)	Midpoint Y (mm)	Midpoint X (mm)	Midpoint Y (mm)	Midpoint X (mm)	Midpoint Y (mm)
1	260.7	188.6	-16.3	189.9	-293.9	183.1
2	261.4	484.5	-16.3	481.9	-293.5	477.3
3	256.5	777.3	-18.6	771.8	-296.5	767.5
4	257.7	1073.7	-19.8	1066.7	-297.4	1062.6
5	255.0	1357.8	-19.8	1356.4	-297.4	1353.9
6	249.2	1652.1	-23.4	1649.8	-301.3	1644.5
7	248.5	1930.6	-25.7	1939.5	-302.3	1932.7

Table B.3: The errors in the distances between the two different colour markers and the distances between two lines.

Dist.	y-axis			Line	x-axis	
					Red-Green	Green-Blue
	Red	Green	Blue	1	276.9	277.7
1-2	295.9	292.1	294.2	2	277.6	277.2
2-3	292.9	289.9	290.3	3	275.2	277.9
3-4	296.3	294.8	295.1	4	277.5	277.6
4-5	284.1	289.7	291.2	5	274.8	277.6
5-6	294.3	293.4	290.7	6	272.5	277.9
6-7	278.6	289.7	288.1	7	274.2	276.5
AVG	290.3	291.6	291.6	AVG	275.5	277.5
Error (%)	1.2			Error (%)	1.5	

C Gait Parameters for Each Gait Trial

Table C.1: The sets of gait parameters for the linear progression gait trials.

Trial	$A_p(^\circ)$	$\psi_p(^\circ)$	ω(rad/s)
1	30	144	π
2	20	144	π
3	30	120	π
4	20	120	π

Table C.2: The sets of gait parameters for the turning gait trials.

Trial	$A_p(^\circ)$	$\psi_p(^\circ)$	$O_y(^\circ)$	ω(rad/s)
1	30	144	10	π
2	20	144	20	π
3	30	120	30	π
4	20	120	10	π

Table C.3: The sets of gait parameters for the rolling gait trials.

Trial	$A(^\circ)$	ω(rad/s)
1	15	π
2	20	π
3	30	π
4	50	π

Table C.4: The sets of gait parameters for the rotating gait trials.

Trial	$A_p(^\circ)$	$A_y(^\circ)$	$\psi_p(^\circ)$	$\psi_y(^\circ)$	ω(rad/s)
1	10	30	90	120	π
2	5	25	90	120	π
3	5	30	90	120	π
4	10	25	90	120	π

Table C.5: The sets of gait parameters for the sidewinding gait trials.

Trial	$A_p(^\circ)$	$A_p(^\circ)$	$\psi(^\circ)$	ω(rad/s)
1	5	25	144	0.5π
2	25	25	144	0.5π
3	5	25	120	0.5π
4	25	25	120	0.5π

D Mean and Deviation of Motion Trajectories

Table D.1: The mean and the deviation of the motion trajectories of the linear progression trials

Trial	Experiment						Simulation					
	r (cm)		θ (°)		ϕ (°)		r (cm)		θ (°)		ϕ (°)	
	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev
1a	3.49	0.01	90.56	1.06	0.1034	0.0056	3.31	0.07	89.92	0.37	-0.0240	0.0000
1b	3.41	0.00	91.20	0.05	0.1579	0.0006	3.14	0.05	-89.50	0.60	0.1010	0.0001
2a	2.24	0.06	90.19	3.27	0.2869	0.0020	2.13	0.05	89.93	0.40	-0.0229	0.0000
2b	2.31	0.05	-90.79	2.49	-0.2524	0.0011	2.60	0.05	-89.33	0.42	0.1213	0.0002
3a	4.16	0.10	90.21	2.55	0.3698	0.0100	4.09	0.08	89.61	0.39	-0.0068	0.0001
3b	4.09	0.11	-91.56	3.24	-0.8190	0.0072	3.96	0.06	-89.24	0.73	0.1869	0.0001
4a	2.51	0.10	90.20	3.53	0.3765	0.0005	2.71	0.04	89.64	0.34	-0.0270	0.0001
4b	2.78	0.07	-92.07	1.46	-0.5122	0.0005	2.59	0.05	-89.06	1.80	0.2359	0.0003

Table D.2: The mean and the deviation of the motion trajectories of the turning trials

Trial	Experiment						Simulation					
	r (cm)		θ (°)		ϕ (°)		r (cm)		θ (°)		ϕ (°)	
	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev
1a	3.492	0.015	72.36	0.19	-2.8223	0.0128	3.221	0.052	70.81	1.37	-2.3125	0.0021
1b	3.392	0.005	-89.10	0.06	2.5287	0.0003	3.229	0.003	-91.24	1.37	2.3210	0.0020
2a	1.564	0.059	85.59	5.81	-1.8489	0.0013	1.551	0.039	84.52	2.76	-2.0804	0.0001
2b	1.608	0.208	-85.26	6.90	2.0396	0.0007	1.401	0.026	-91.98	5.93	1.7020	0.0002
3a	4.219	0.724	62.82	6.19	-8.1003	0.0033	3.971	1.019	54.20	6.25	-9.0990	0.0034
3b	4.305	0.159	-68.69	15.34	8.5179	0.0034	3.771	0.138	-71.21	15.26	9.1591	0.0016
4a	1.828	0.050	86.57	4.17	-1.1012	0.0008	2.023	0.041	86.66	2.43	-1.4559	0.0001
4b	2.000	0.043	-87.66	2.68	0.8940	0.0008	1.957	0.036	-87.03	2.87	1.4023	0.0001

Table D.3: The mean and the deviation of the motion trajectories of the rolling trials

Trial	Experiment						Simulation					
	r (cm)		θ (°)		ϕ (°)		r (cm)		θ (°)		ϕ (°)	
	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev
1a	17.40	0.66	-88.85	1.17	-0.950	0.023	14.82	0.01	-89.82	0.36	-1.557	0.006
1b	16.73	0.26	91.31	0.25	0.982	0.055	14.09	0.01	75.84	0.03	0.697	0.001
2a	17.49	0.89	-98.25	4.15	-2.268	0.016	15.13	0.42	-92.61	2.27	-1.245	0.003
2b	17.43	0.24	94.03	5.74	2.833	0.017	13.21	0.38	90.55	0.77	0.186	0.000
3a	19.74	0.85	-105.64	0.80	-0.439	0.020	15.73	0.02	-105.69	0.15	-1.338	0.004
3b	19.66	0.43	77.39	0.77	0.561	0.043	14.09	0.01	75.84	0.03	0.697	0.001
4a	11.40	0.17	-83.76	1.99	1.670	0.006	10.96	0.55	-96.34	2.85	-3.255	0.019
4b	10.79	0.79	93.54	6.35	-1.686	0.017	8.37	0.13	99.99	3.65	0.978	0.003

Table D.4: The mean and the deviation of the motion trajectories of the rotating trials

Trial	Experiment						Simulation					
	r (cm)		θ (°)		ϕ (°)		r (cm)		θ (°)		ϕ (°)	
	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev
1a	1.07	0.16	134.61	13.59	-12.272	0.018	1.21	0.37	81.99	0.15	-17.830	0.019
1b	0.91	0.50	82.92	1.09	20.847	0.019	0.65	0.06	99.87	1.89	18.378	0.009
2a	0.73	0.07	148.80	23.89	-6.779	0.016	0.40	0.14	129.96	35.93	-13.682	0.004
2b	0.84	0.36	58.59	7.76	7.760	0.011	0.83	0.07	-4.79	21.93	14.196	0.017
3a	0.75	1.37	136.06	72.24	-7.619	0.020	0.69	0.06	120.11	21.45	-14.603	0.004
3b	0.98	1.08	48.05	94.52	7.900	0.010	0.83	0.35	25.00	25.29	16.627	0.014
4a	0.78	0.38	52.64	2.94	-11.941	0.014	0.46	0.26	53.92	11.91	-15.580	0.037
4b	1.47	0.09	109.70	19.50	9.954	0.009	0.58	0.52	145.84	193.32	17.588	0.004

Table D.5: The mean and the deviation of the motion trajectories of the sidewinding trials

Trial	Experiment						Simulation					
	r (cm)		θ (°)		ϕ (°)		r (cm)		θ (°)		ϕ (°)	
	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev	Average	Dev
1a	6.20	0.05	-83.37	3.80	1.966	0.001	3.65	0.06	-77.91	0.99	-0.371	0.001
1b	6.34	0.19	88.08	1.90	-1.350	0.004	2.84	0.03	104.47	1.85	0.889	0.000
2a	6.61	0.09	-69.06	1.36	0.499	0.026	4.52	0.01	-49.76	1.01	-0.373	0.000
2b	6.43	1.31	109.97	7.45	0.256	0.020	3.51	0.63	130.53	18.73	0.821	0.022
3a	3.64	1.52	-85.18	3.90	-0.554	0.015	5.77	0.07	-79.09	3.51	1.941	0.003
3b	4.44	0.20	100.20	2.01	0.040	0.003	5.26	0.17	87.38	3.46	-1.744	0.002
4a	5.04	0.12	-60.55	2.86	-0.790	0.006	6.86	2.51	-74.83	11.00	0.344	0.067
4b	5.19	0.06	129.02	4.47	2.164	0.010	6.37	3.12	97.98	13.82	0.613	0.083

E Deviation Ratios of Motion Trajectories

Table E.1: The deviation ratios of the linear progression gait trials.

Trial	Experiment	Simulation
	$h_r(\%)$	$h_r(\%)$
1a	0.31	2.08
1b	0.06	1.54
2a	2.79	2.39
2b	2.11	1.98
3a	2.46	1.97
3b	2.57	1.42
4a	4.01	1.39
4b	2.43	1.81
Avg.	2.09	1.82

Table E.2: The deviation ratios of the turning gait trials.

Trial	Experiment			Simulation		
	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$
1a	0.43	0.27	0.45	1.60	1.94	0.09
1b	0.15	0.07	0.01	0.09	1.51	0.08
2a	3.80	6.79	0.07	2.53	3.26	0.00
2b	12.97	8.09	0.03	1.85	6.44	0.01
3a	17.17	9.85	0.04	25.65	11.53	0.04
3b	3.69	22.33	0.04	3.65	21.43	0.02
4a	2.71	4.82	0.07	2.02	2.80	0.00
4b	2.15	3.06	0.09	1.83	3.29	0.00
Avg.	5.38	6.91	0.10	4.90	6.53	0.03

Table E.3: The deviation ratios of the rolling gait trials.

Trial	Experiment			Simulation		
	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$
1a	3.81	1.31	2.46	0.09	0.40	0.38
1b	1.53	0.28	5.60	0.04	0.04	0.19
2a	5.09	4.22	0.72	2.80	2.45	0.22
2b	1.40	6.11	0.59	2.87	0.85	0.21
3a	4.32	0.76	4.50	0.12	0.15	0.30
3b	2.21	0.99	7.64	0.04	0.04	0.19
4a	1.51	2.38	0.35	5.03	2.96	0.57
4b	7.35	6.79	1.00	1.54	3.65	0.35
Avg.	3.40	2.86	2.86	1.57	1.32	0.30

Table E.4: The deviation ratios of the rotating gait trials.

Trial	Experiment			Simulation		
	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$
1a	14.9	10.1	0.1	30.21	0.2	0.1
1b	54.6	1.3	0.1	9.64	1.9	0.1
2a	9.4	16.1	0.2	35.01	27.6	0.0
2b	42.9	13.3	0.1	8.58	457.9	0.1
3a	182.1	53.1	0.3	9.16	17.9	0.0
3b	111.1	196.7	0.1	42.50	101.2	0.1
4a	47.9	5.6	0.1	57.39	22.1	0.2
4b	6.1	17.8	0.1	89.60	132.6	0.0
Avg.	58.61	39.24	0.15	35.26	95.16	0.08

Table E.5: The deviation ratios of the sidewinding gait trials.

Trial	Experiment			Simulation		
	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$	$h_r(\%)$	$h_\theta(\%)$	$h_\phi(\%)$
1a	0.88	4.56	0.07	1.65	1.27	0.22
1b	2.96	2.16	0.26	1.00	1.77	0.03
2a	1.42	1.97	5.20	0.20	2.03	0.13
2b	20.28	6.77	7.95	18.03	14.35	2.65
3a	41.76	4.58	2.78	1.15	4.44	0.17
3b	4.52	2.01	7.35	3.30	3.96	0.10
4a	2.34	4.72	0.77	36.58	14.70	19.52
4b	1.24	3.46	0.48	49.00	14.10	13.53
Avg.	9.42	3.78	3.11	13.87	7.08	4.54

F Data of Motion Trajectories

Table F.1: The final positions of the motion trajectories of the linear progression gait trials.

Trial	Experiment	Model	Simulation
	r(cm)	r(cm)	r(cm)
1a	20.95	20.85	19.99
1b	20.47	20.85	20.02
2a	13.46	12.66	12.79
2b	13.88	12.70	15.58
3a	24.96	23.03	24.55
3b	24.54	23.87	23.74
4a	15.09	14.34	16.29
4b	16.68	14.90	15.57

Table F.2: The final positions of the motion trajectories of the turning gait trials.

Trial	Experiment			Model			Simulation		
	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$
1b	20.3	-83.9	67.4	21.1	-93.8	91.7	19.2	-85.1	80.6
1b	20.28	83.94	67.44	21.05	93.81	91.74	19.20	85.15	80.58
2a	9.38	94.41	118.75	10.46	93.57	117.11	9.30	99.00	122.20
2b	9.65	-93.89	96.19	10.34	-93.59	102.66	8.40	-88.02	99.51
3a	25.31	117.18	165.06	25.50	102.62	145.29	23.83	119.08	174.32
3b	25.83	-79.60	97.02	25.28	-102.67	94.32	22.62	-61.65	118.05
4a	10.97	93.43	104.79	14.33	92.61	105.42	12.14	93.35	108.65
4b	12.00	-86.56	98.71	14.01	-92.49	95.12	11.74	-92.97	91.53

Table F.3: The final positions of the motion trajectories of the rolling gait trials.

Trial	Experiment			Model			Simulation		
	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$
1a	104.36	-90.88	-36.76	115.65	-86.71	-7.63	88.80	-86.86	-23.97
1b	100.40	91.94	-24.60	115.65	85.98	-22.37	84.52	96.25	-11.65
2a	104.58	-94.03	-38.47	109.63	-91.02	-21.85	90.79	-91.46	-27.36
2b	104.95	98.25	-6.60	109.63	90.82	-18.14	79.27	92.70	-18.83
3a	118.45	-91.15	-18.31	92.79	-92.43	-35.09	94.30	-89.29	-12.10
3b	118.00	94.18	-11.08	92.79	92.62	-24.91	84.52	96.25	-11.65
4a	64.73	-80.79	-39.19	48.54	-88.30	-50.38	65.77	-96.34	-69.44
4b	68.37	83.76	-60.63	48.54	95.41	-42.93	50.21	99.99	-44.01

Table F.4: The final positions of the motion trajectories of the rotating gait trials.

Trial	Experiment			Model			Simulation		
	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$
1a	6.19	101.14	-73.63	1.19	-129.07	-89.14	6.31	22.73	-106.99
1b	3.98	143.22	125.08	0.68	-18.31	129.64	3.24	157.86	110.25
2a	4.35	148.80	-40.68	0.59	-105.82	-38.18	2.41	129.96	-82.09
2b	5.02	58.59	46.56	0.61	108.91	41.26	4.95	-4.79	85.17
3a	4.50	136.06	-45.71	0.67	-111.80	-47.20	4.15	120.11	-87.62
3b	5.86	48.05	47.40	0.70	113.12	48.05	4.99	25.00	99.76
4a	4.70	52.64	-71.65	1.13	-119.39	-75.07	2.77	53.92	-93.48
4b	8.84	109.70	59.72	1.16	125.16	77.57	3.47	145.84	105.53

Table F.5: The final positions of the motion trajectories of the sidewinding gait trials.

Trial	Experiment			Model			Simulation		
	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$	r(cm)	$\theta(^\circ)$	$\phi(^\circ)$
1a	37.22	-83.37	11.80	18.19	-80.29	-3.12	21.92	-77.91	-2.22
1b	38.02	88.08	-8.10	17.83	104.37	10.14	17.02	104.47	5.33
2a	39.67	-69.06	3.00	12.54	-71.47	3.17	27.14	-49.76	-2.24
2b	38.61	109.97	1.54	13.44	104.84	-5.58	21.08	130.53	4.93
3a	21.82	-85.18	-3.33	32.94	-90.18	-21.41	34.63	-79.09	11.64
3b	26.64	100.20	0.24	32.77	118.66	37.24	31.58	87.38	-10.46
4a	30.24	-60.55	-4.74	30.40	-88.21	-22.76	41.17	-74.83	2.06
4b	31.13	129.02	12.98	30.43	115.17	23.41	38.22	97.98	3.68

G Errors in Motion Trajectories

Table G.1: The errors in the motion trajectories of the linear progression gait trials.

Trial	Model	Simulation
	r (%)	r (%)
1a	(-)0.5	(-)4.6
1b	1.8	(-)2.2
2a	(-)6.0	(-)5.0
2b	(-)8.5	12.2
3a	(-)7.7	(-)1.7
3b	(-)2.7	(-)3.3
4a	(-)4.9	7.9
4b	(-)10.6	(-)6.7
Avg.	5.4	5.4
Dev.	3.3	3.2

Table G.2: The errors in the motion trajectories of the turning gait trials.

Trial	Model			Simulation		
	r(%)	θ(%)	ϕ(%)	r(%)	θ(%)	ϕ(%)
1a	6.5	17.2	(-)10.4	(-)8.0	(-)2.6	15.0
1b	5.1	(-)12.0	7.8	(-)5.3	(-)1.4	19.5
2a	11.5	(-)0.9	(-)1.4	(-)0.8	4.9	2.9
2b	7.2	0.3	6.7	(-)12.9	6.3	3.5
3a	0.7	(-)12.4	(-)12.0	(-)5.9	1.6	5.6
3b	(-)2.1	(-)29.0	(-)2.8	(-)12.4	22.6	21.7
4a	30.6	(-)0.9	0.6	10.7	(-)0.1	3.7
4b	16.7	(-)6.9	(-)3.6	(-)2.2	(-)7.4	7.3
Avg.	10.1	9.9	5.7	7.3	5.9	9.9
Dev.	9.1	9.3	3.9	4.2	6.7	7.2

Table G.3: The errors in the motion trajectories of the rolling gait trials.

Trial	Model			Simulation		
	r(%)	$\theta(^\circ)$	ϕ(%)	r(%)	$\theta(^\%)$	ϕ(%)
1a	10.8	2.3	16.2	(-)14.9	2.2	7.1
1b	15.2	(-)3.3	1.2	(-)15.8	2.4	7.2
2a	4.8	1.7	9.2	(-)13.2	1.4	6.2
2b	4.5	(-)4.1	(-)6.4	(-)24.5	(-)3.1	(-)6.8
3a	(-)21.7	(-)0.7	(-)9.3	(-)20.4	1.0	3.5
3b	(-)21.4	(-)0.9	(-)7.7	(-)28.4	1.1	(-)0.3
4a	(-)25.0	(-)4.2	(-)6.2	1.6	(-)8.6	(-)16.8
4b	(-)29.0	6.5	9.8	(-)26.6	9.0	9.2
Avg.	16.5	3.0	8.3	18.2	3.6	7.1
Dev.	8.6	1.8	3.9	8.2	3.1	4.5

Table G.4: The errors in the motion trajectories of the rotating gait trials.

Trial	Model		Simulation	
	r(%)	ϕ(%)	r(%)	ϕ(%)
1a	7.3	21.1	9.3	45.3
1b	4.8	3.6	4.8	(-)11.9
2a	5.5	(-)6.1	3.5	101.8
2b	6.5	(-)11.4	7.3	82.9
3a	5.6	3.3	6.1	91.7
3b	7.6	1.4	7.3	110.5
4a	5.2	4.8	4.1	30.5
4b	11.3	29.9	5.1	76.7
Avg.	6.7	10.2	5.9	68.9
Dev.	2.1	10.2	1.9	35.6
Max.	8.81	20.34	7.86	104.51

Table G.5: The errors in the motion trajectories of the sidewinding gait trials.

Trial	Model			Simulation		
	r(%)	θ(%)	ϕ(%)	r(%)	θ(%)	ϕ(%)
1a	(-)51.1	1.7	(-)8.3	(-)41.1	3.0	(-)7.8
1b	(-)53.1	9.1	10.1	(-)55.2	9.1	7.5
2a	(-)68.4	(-)1.3	0.1	(-)31.6	10.7	(-)2.9
2b	(-)65.2	(-)2.9	(-)4.0	(-)45.4	11.4	1.9
3a	50.9	(-)2.8	(-)10.0	58.7	3.4	8.3
3b	23.0	10.3	20.6	18.5	(-)7.1	(-)5.9
4a	0.5	(-)15.4	(-)10.0	36.1	(-)7.9	3.8
4b	(-)2.3	(-)7.7	5.8	22.8	(-)17.2	(-)5.2
Avg.	39.3	6.4	8.6	38.7	8.7	5.4
Dev.	25.3	4.7	5.6	13.4	4.3	2.2